



Основные характеристики, описываемые в CSS

CSS придаёт HTML-страницам внешний вид: цвета, шрифты, размеры, отступы, рамки, тени, размещение и адаптивность.



Что такое CSS и почему он нужен на уроке информатики

CSS (каскадные таблицы стилей) появился в 1996 году, чтобы отделить структуру от оформления. HTML задаёт содержание, а CSS — внешний вид. Один набор правил может применяться на множестве страниц, делая дизайн единообразным и удобным для правок.



Цвет текста, фона и изображение в оформлении страницы

.1

Свойство `color` управляет цветом текста, поэтому заголовки и важные фразы можно выделять контрастом. Это помогает лучше ориентироваться на странице и читать без лишних усилий.

.2

Свойство `background-color` задаёт цвет подложки элемента. А `background-image` подключает картинку фона, превращая обычные блоки в заметные секции — например, «баннер» или карточку с новостями.

.3

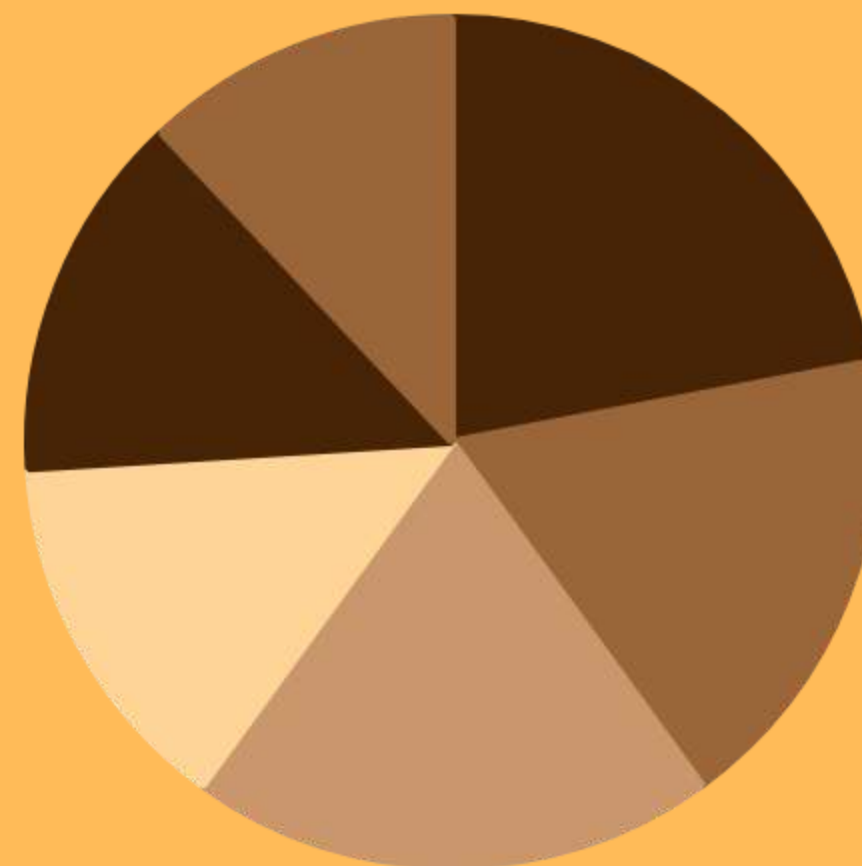
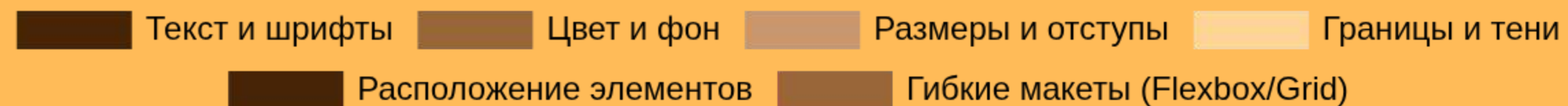
Для фона настраивают повторение, положение и размер. Благодаря этим параметрам изображение не «ломает» макет и поддерживает стиль: кнопки, блоки с информацией и разделители выглядят аккуратно и целостно.

Основные группы CSS-свойств

CSS влияет сразу на три стороны страницы: что видно (содержимое), как это выглядит (оформление) и где это находится (размещение).

Самый большой вклад обычно дают шрифты и размеры: именно они сильнее всего меняют читаемость и общий «ритм» макета.

Учебная обобщённая структура тем курса (разделение групп свойств CSS). Доли приведены как ориентиры для наглядности.



Шрифты и оформление текста

.1

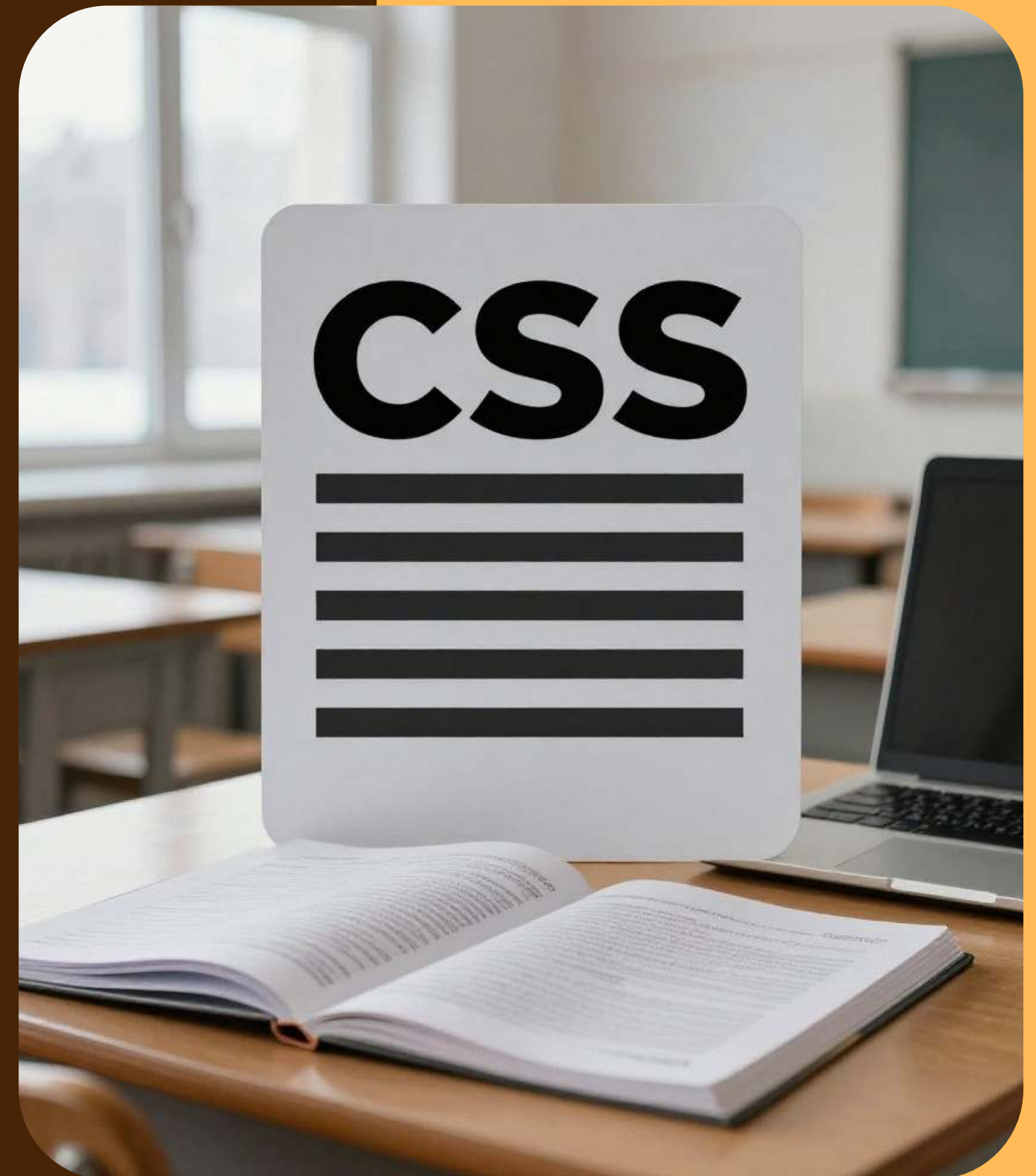
Как CSS задаёт вид текста

CSS управляет `font-family`, `font-size`, `font-weight`, `font-style`, `line-height` и `text-align`. Благодаря этим свойствам можно выбрать семейство шрифта, размер букв, жирность, наклон, высоту строк и выравнивание текста по смыслу.

.2

Зачем это нужно на практике

Разница между заголовком и абзацем хорошо видна: заголовок обычно крупнее и жирнее, а в абзаце важнее комфорт чтения за счёт `line-height` и выравнивания. Так структура страницы становится понятнее с первого взгляда.



Свойства текста и их назначение

Эти свойства помогают сделать текст читабельным, структурированным и аккуратным: от размера и толщины шрифта до интервалов между буквами.

Точность настройки шрифта и интервалов напрямую влияет на читабельность: даже небольшие изменения делают текст либо комфортным, либо утомительным для глаз.

Базовые разделы CSS Fonts и CSS Text.

Свойство	Что задаёт
font-size	размер шрифта
font-weight	толщина начертания
line-height	межстрочное расстояние
text-align	выравнивание
text-decoration	оформление линиями
letter-spacing	интервал между буквами

6 ТИПОВ

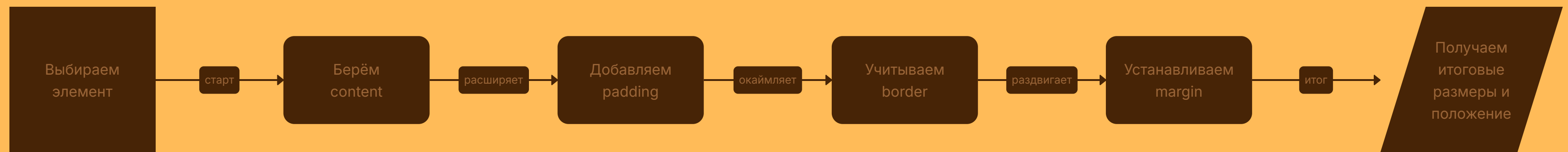
Используя разные единицы измерения, можно гибко подстраивать размеры под разные экраны и задачи.

Учебный материал по единицам измерения CSS (px, %, em, rem, vw, vh) и правилам width/height, min-/max-.

В CSS ширину и высоту задают width и height, а ограничивают через min/max. Для значений используют пиксели, проценты, em, rem, vw и vh — они по-разному ведут себя на разных устройствах.

Модель коробки: из чего состоит блок

Схема поясняет, как браузер учитывает части элемента при расчёте размеров и расстояний.



Границы, скругления и тени

.1

Свойство `border` задаёт границу элемента, а `border-width`, `border-style` и `border-color` уточняют толщину, тип линии и цвет. Так карточки и кнопки становятся хорошо различимыми на фоне страницы.

.2

Скругления создают `border-radius`: углы выглядят мягче и современнее, а интерфейс — аккуратнее. Это особенно заметно на элементах вроде меню и панелей с информацией.

.3

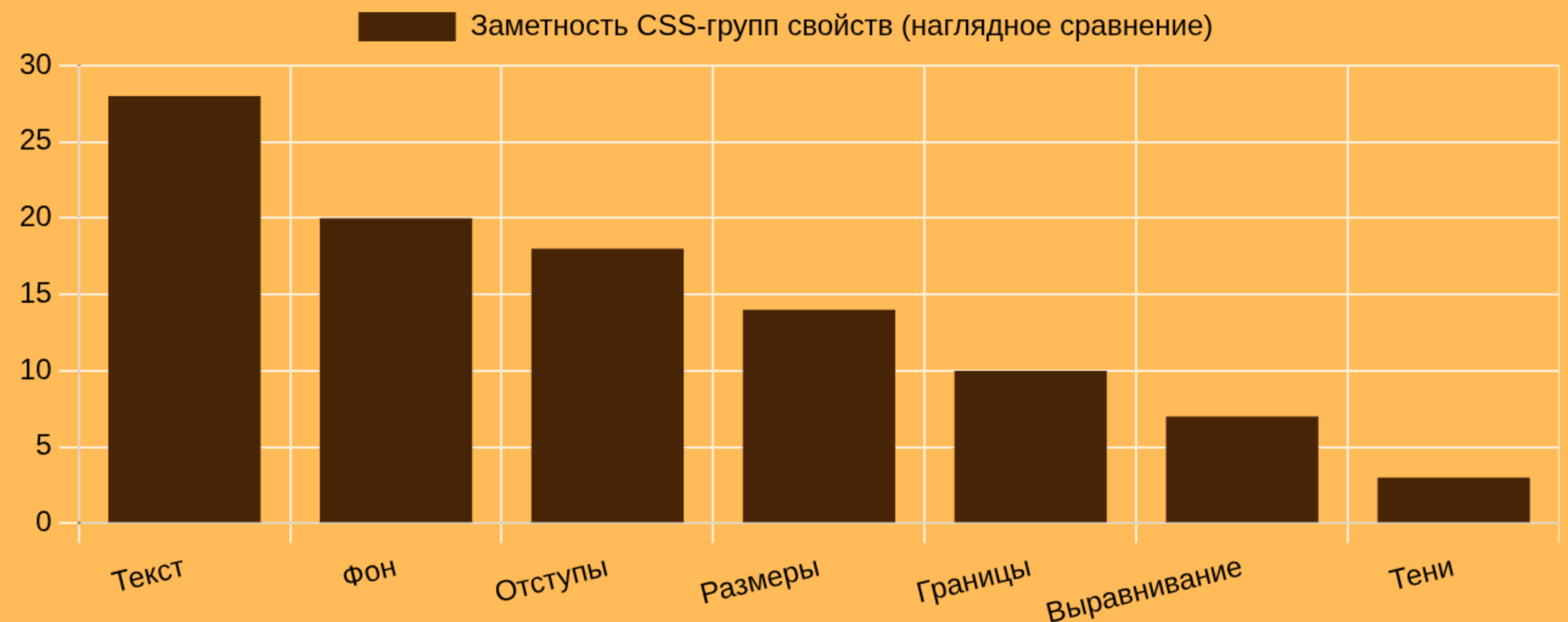
Тени задаёт `box-shadow`. Они визуально отделяют блоки, помогают выделять главные элементы и создают ощущение глубины, не перегружая дизайн.

Какие свойства чаще всего заметны на странице

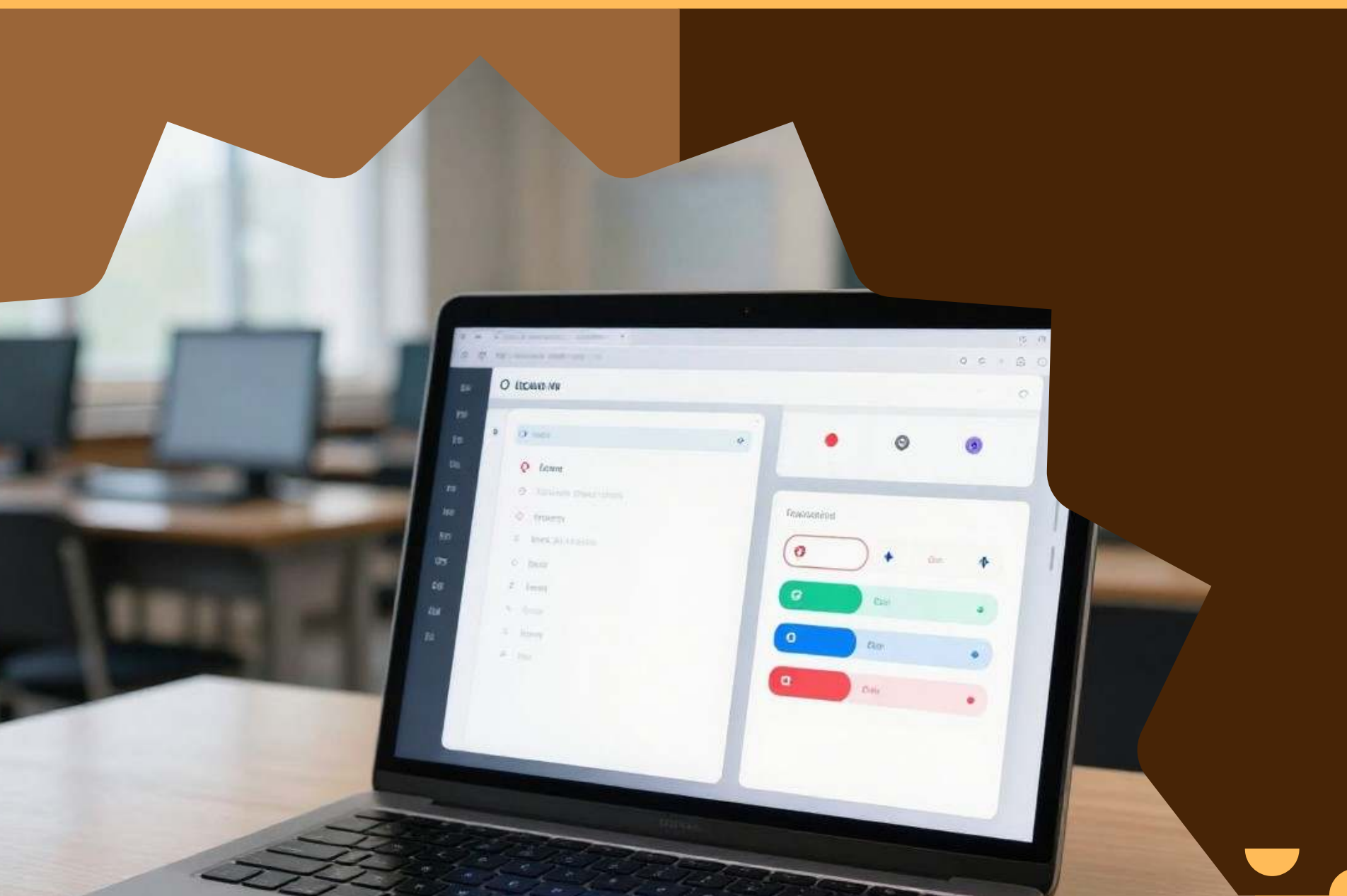
Когда меняется текст, фон или расстояния, человек сразу замечает изменения, даже если остальная вёрстка не тронута.

Диаграмма подчёркивает: небольшие правки в CSS способны заметно изменить восприятие макета — особенно через читаемость и контраст.

Учебная оценка заметности свойств CSS в интерфейсе (ориентировочные веса для наглядности, а не результаты измерений).



Списки, ссылки и элементы навигации



- Списки оформляются с помощью list-style: можно менять маркеры, их положение и внешний вид. Это помогает превращать нумерацию и «точки» в понятную структуру для меню, чек-листов и инструкций на странице.

- Ссылки стилизуются через color и text-decoration, а также через состояния :hover, :visited, :active и :focus. Когда оформление меняется при наведении и выборе, пользователю проще понять, где находится интерактивный элемент.

Классы, идентификаторы и селекторы

.1

Стили можно применять по-разному: по тегу, по классу или по идентификатору. Селекторы позволяют выбрать именно те элементы, которым нужны изменения внешнего вида, даже если на странице много похожих блоков.

.2

Класс обычно используют для нескольких элементов: например, одни и те же стили для всех карточек товара. Идентификатор чаще задают одному элементу, поэтому он помогает назначать уникальное оформление конкретному блоку на странице.

.3

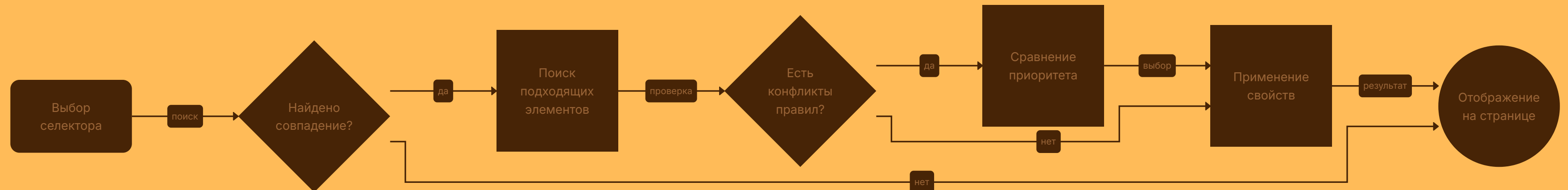
Селекторы по вложенности учитывают структуру документа: стиль применяется, когда элемент находится внутри определённого родителя. Это удобно для меню, где разные уровни вложенности можно оформить по-разному.

.4

Селекторы по состоянию (например, когда элемент активен или доступен для фокуса) меняют поведение визуально. Точность выбора влияет на то, какие правила сработают в итоге, особенно в больших проектах.

Как CSS-правило применяется к элементу

Алгоритм выбора и применения стилей: от селектора до финального отображения.



.1

Свойство `position` управляет тем, как элемент размещается относительно документа и других объектов: `static`, `relative`, `absolute`, `fixed` и `sticky`. Выбор режима определяет, входит ли блок в обычный поток страницы.

.2

Параметры `top`, `right`, `bottom` и `left` задают смещение. Они особенно важны вместе с относительным и абсолютным позиционированием: можно «сдвинуть» элемент от исходного положения или заданной точки.

.3

При `static` элемент ведёт себя как обычно в потоке документа. При `relative` смещение происходит относительно того места, где элемент располагался бы без позиционирования, поэтому контекст потока сохраняется частично.

.4

`absolute` выводит элемент из нормального потока, поэтому остальные блоки ведут себя так, будто его нет; `fixed` закрепляет его относительно окна браузера, а `sticky` сочетает обычное поведение и «прилипание» при прокрутке.

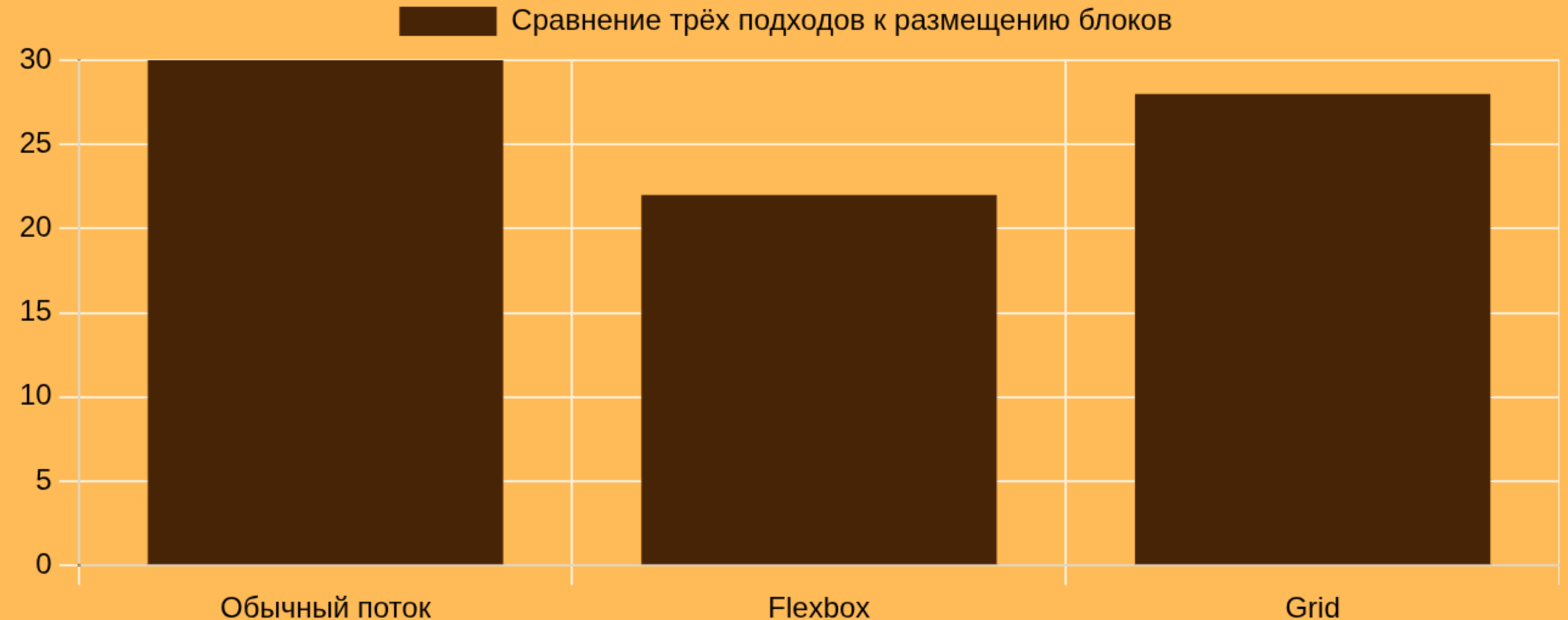
Позиционирование элементов на странице

Способы расположения блоков на веб-странице

В обычном потоке элементы идут последовательно: текст и простые блоки легче всего строить именно так. Flexbox ориентирован на одну ось, а Grid — на две, когда важны и строки, и колонки.

Выбор инструмента зависит от задачи: для линейных списков и простых блоков хватает потока, а для сложных макетов быстрее и точнее использовать Flexbox или Grid.

Обобщение практик в CSS-макетах: обычный поток документа, Flexbox и Grid (без привязки к конкретным числам).



Flexbox и Grid в современных макетах



- Flexbox удобен, когда нужно распределить элементы в одном направлении и быстро выровнять их по главной или поперечной оси. Поэтому его часто применяют для меню, панелей кнопок и карточек с одинаковой логикой расположения.

- Grid создаёт двухмерную сетку, где одновременно задаются строки и колонки. Он подходит для страниц с несколькими секциями и «витринной» структурой; свойства вроде `grid-template-columns` и `gap` ускоряют сборку макета и делают его аккуратнее.

Адаптивность и медиа-запросы



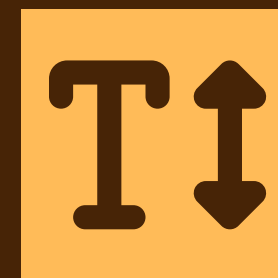
Разные размеры экранов

Медиа-запросы @media меняют стили в зависимости от ширины и других параметров устройства. Благодаря этому шрифты и блоки подстраиваются, чтобы страница оставалась удобной на телефонах и планшетах.



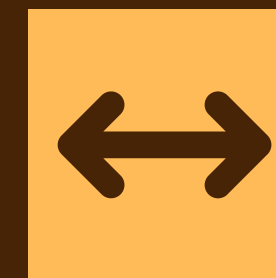
Гибкая сетка и отступы

При адаптивной вёрстке меняются количество колонок, ширина элементов и расстояния между ними. Так можно сохранять читабельные интервалы, не «уплотняя» текст при уменьшении экрана.



Управление читаемостью

На разных устройствах важны не только размеры, но и плотность текста. Через @media можно увеличить line-height или изменить font-size, чтобы абзацы выглядели аккуратно и не утомляли при чтении.



Ориентация и условия

@media реагирует не только на размер, но и на ориентацию (портрет/ландшафт) и другие условия. Это помогает подобрать оптимальный вид интерфейса под сценарий использования пользователя.

Сравнение часто используемых свойств CSS

Эти свойства отвечают за ключевые части оформления: цвет, отступы, рамки, способ отображения и позиционирование. Понимание назначения каждого пункта упрощает создание страницы и отладку стилей.

Комбинируя свойства, можно по частям собирать внешний вид блока: сначала фон и текст, затем расстояния и рамки, а в конце — как элемент отображается и где он находится на странице.

Справочные разделы CSS2.1 и актуальные спецификации CSS.

Свойство	Назначение
color	Цвет текста
background-color	Цвет фона
margin	Внешние отступы
padding	Внутренние отступы
border	Рамка
display	Отображение элемента
position	Размещение элемента

Каскад, наследование и приоритет

.1

CSS использует каскад: если несколько правил могут подойти одному элементу, решающим становится порядок и приоритет. Более точные селекторы и специальные правила способны переопределить «общие» настройки.

.2

Наследование позволяет части свойств переходить от родителя к вложенным элементам. Например, цвет текста или семейство шрифта часто задают на уровне контейнера, чтобы не дублировать стили для каждого тега.

.3

При конфликте учитываются специфичность селектора, последовательность записи и значение `!important`. Поэтому два одинаковых элемента могут выглядеть по-разному, если для них срабатывают разные правила в каскаде.

CSS описывает внешний вид — и помогает сделать сайт удобным

CSS задаёт цвет, шрифты, размеры, отступы, границы, тени, расположение и адаптивное поведение страницы. Благодаря этому сайт становится читаемым и аккуратным на разных экранах, а модель коробки, селекторы, каскад, Flexbox и Grid дают основу уверенной вёрстки.