

Учебник для пенсионеров

# **СТИЛИ CSS**

## **в веб-разработке**

Екатеринбург  
2025

# ОГЛАВЛЕНИЕ

## Введение: Добро пожаловать в мир творчества!

- Кому и зачем нужна эта книга?
  - Почему CSS – это не страшно, а очень увлекательно? (Сравнение с: ремонтом квартиры (HTML – стены, CSS – обои и расстановка мебели), вязанием, садоводством).
  - Что нам понадобится для начала? (Только компьютер, браузер и простейший блокнот. Никаких сложных программ!).
  - Важное правило: Не бояться экспериментировать и ошибаться! У нас всё безопасно.
- 

## Часть 1: Первые шаги. Знакомимся с кистями и красками

- **Глава 1:** Что такое CSS? Проще простого!
  - История в картинках: Страница без CSS (голые стены) → Страница с CSS (уютный дом).
  - Где живут стили? Три способа «прицепить» стиль к странице:
    1. Рядом с элементом (как записка на холодильник) – inline-стили.
    2. В «шапке» страницы (как инструкция для всей комнаты) – стили в `<head>`.
    3. В отдельной «книге рецептов» (самый удобный способ!) – внешний файл `.css`.

- **Глава 2:** Наш первый волшебный ингредиент – ЦВЕТ.

- Как говорить с компьютером о цвете? Простые слова (red, blue) и «волшебные коды» (#FF0000, rgb(255,0,0)).
  - Практикум №1: Раскрашиваем заголовок и абзац. Видим результат мгновенно!
  - Цвет текста (color) vs цвет фона (background-color). Пробуем оба.

- **Глава 3:** Учимся управлять ТЕКСТОМ.

- Выбираем шрифт (font-family): Как найти красивые и понятные шрифты?
  - Размер букв (font-size): Почему px, em и % — как сантиметры и дюймы в шитье.
  - Жирность (font-weight) и наклон (font-style): Выделяем важное.
  - Выравнивание (text-align): Как в текстовом редакторе Word.

- Практикум №2: Пишем красивое поздравительное письмо в веб-формате.
- 

## Часть 2: Осваиваем пространство. Расставляем мебель в комнате

### ● Глава 4: Загадочный БЛОК. Основа основ.

- Что такое блочный элемент? Каждый блок – как отдельная рамка для фотографии.
- Волшебные отступы: `margin` (расстояние между рамками) и `padding` (поле внутри рамки, вокруг фото). Объяснение с помощью реальной фоторамки.
- Практикум №3: Создаем свою первую визитку или открытку с отступами.

### ● Глава 5: Рамки и тени – создаем настроение.

- Свойство `border`: сплошная, пунктирная, цвет, толщина. Делаем аккуратную рамку.
- Свойство `box-shadow`: добавляем легкую тень, чтобы элемент «парил» над страницей.
- Практикум №4: Украшаем нашу визитку красивой рамкой и тенью.

### ● Глава 6: Как управлять размером «рамки»?

- Ширина (`width`) и высота (`height`): в пикселях и процентах.
  - Важное понятие: `max-width` – чтобы наша «мебель» влезала в любой «дверной проем» (размер экрана).
- 

## Часть 3: Магия превращений. Делаем страницу живой и удобной

### ● Глава 7: «Умный» способ применять стили – КЛАССЫ.

- Зачем плодить одинаковые инструкции? Даем элементам «имена» (классы).
- Как создать класс в CSS (точка перед названием) и «присвоить» его в HTML.
- Практикум №5: Создаем стиль для «особо важного объявления» и «изящной подписи».

### ● Глава 8: Простое меню навигации – гордость любого сайта.

- Создаем список ссылок (<ul> и <li>).
- Превращаем скучный список в красивую горизонтальную панель с помощью CSS (убираем маркеры, делаем элементы в строку, добавляем фон и отступы).

### ● Глава 9: Чудеса при **наведении мышки**.

- Псевдокласс :hover – что это такое?
  - Практикум №6: Делаем так, чтобы кнопки меню меняли цвет при наведении, а ссылки – подчеркивались.
- 

## Часть 4: Финальные штрихи и большие возможности

### ● Глава 10: Работа с **ИЗОБРАЖЕНИЯМИ**.

- Как сделать изображение адаптивным (max-width: 100%)?
- Простая рамка и скругление углов (border-radius) – чтобы фото выглядело современно.
- Как добавить подпись к фотографии?

### ● Глава 11: Секрет гибких страниц – **CSS Flexbox** (объясняем без страшных слов!).

- Простая аналогия: Контейнер (ящик для инструментов) и гибкие элементы (инструменты внутри).
- Всего одно свойство display: flex для контейнера – и элементы выстраиваются в ряд или столбик, становятся гибкими!
- Практикум №7: Создаем простую галерею фотографий, которая аккуратно выстраивается сама.

### ● Глава 12: Как сделать так, чтобы сайтом было удобно пользоваться и на телефоне? (**Адаптивность**).

- Простое объяснение «медиа-запросов» (@media): Это как разные инструкции для большого телевизора и маленького смартфона.
  - Пример: Меняем размер шрифта и расположение элементов на маленьком экране.
- 

## Заключение: Ваше творчество только начинается!

- Подводим итоги: Что мы теперь умеем?
- Где искать вдохновение? (Советуем простые и красивые сайты).
- Куда двигаться дальше? (Добрые рекомендации для следующего шага).

- Напутствие: Вы теперь не просто пользователи, а творцы цифрового пространства!
- 

## Приложения (полезные шпаргалки):

1. **Мини-словарик:** Перевод самых частых терминов с «компьютерного» на русский.
  2. **Палитра цветов:** Таблица с красивыми и понятными названиями цветов и их кодами.
  3. **Сборник готовых «рецептов»:** Код для кнопки, красивой цитаты, рамки для фото – чтобы просто скопировать и использовать.
  4. **Список добрых и бесплатных ресурсов:** Где брать шрифты, иконки, картинки.
- 

## Философия этого оглавления:

- **Метафоры из жизни:** Ремонт, садоводство, рукоделие.
- **Постепенность:** От цвета и текста — к блокам и макетам.
- **Мгновенный результат:** После каждой теоретической главы — практикум с видимым, осозаемым итогом.
- **Акцент на нужное:** Не все свойства CSS, а только те, что дают максимальный визуальный эффект и понимание.
- **Уважение и поддержка:** Тон дружеской беседы, а не лекции.

# **Введение: Добро пожаловать в мир творчества!**

## **● Кому и зачем нужна эта книга?**

Дорогой друг!

Перед тем как мы откроем дверь в удивительный мир веб-дизайна, давайте честно ответим на самый главный вопрос: **«А мне-то это зачем?»**

Эта книга написана специально **для вас**. Не для программистов, не для студентов, а для людей мудрых, опытных и любознательных, чья основная жизнь прошла вдали от компьютерных терминов и цифровых технологий.

---

## **Кому эта книга будет верным помощником?**

### **1. Тем, кто хочет наконец «подружиться» с компьютером не на уровне пользователя, а на уровне творца.**

Вы уже умеете писать письма по электронной почте, искать информацию в интернете и общаться в видеозвонках. Но всегда было интересно: а как ЭТО сделано? Как создаются эти красивые странички с поздравлениями, фотоальбомами, рассказами о внуках? Эта книга даст ответ и инструменты.

### **2. Тем, у кого есть хобби, и кто хочет рассказать о нем миру красиво.**

Вы вяжете потрясающие скатерти, пишете мемуары, разводите редкие цветы на даче, коллекционируете марки или печете фирменные пироги. Интернет — это огромная витрина. С помощью CSS вы сможете создать не просто страничку с текстом, а по-настоящему красивый, уютный уголок в сети, который будет отражать вашу душу и мастерство.

### **3. Тем, кто хочет поддерживать связь с семьей на современном, понятном младшим поколениям языке.**

Представьте, что вы не просто отправляете детям и внукам архив фотографий по почте, а даете им ссылку на вашу **персональную интернет-газету** с историями, фотоальбомами и объявлениями. Или создаете красивый сайт-открытку ко дню рождения правнука. Это невероятно сближает и вызывает искреннее восхищение: «Бабушка, да ты крутая!»

#### **4. Тем, кто считает, что «мозги должны работать», и ищет новую гимнастику для ума.**

Изучение CSS — это не зубрежка. Это **логика, внимание к деталям и творчество одновременно**. Это как разгадывать кроссворд, собирать пазл и рисовать картину — все в одном. Это идеальная тренировка для поддержания когнитивных способностей в тонусе.

#### **5. Тем, кто немного побаивается технологий, но чувствует, что за ними — будущее.**

Мы будем двигаться маленькими, очень уверенными шагами. Никакой воды, никакой сложной теории «про запас». Только то, что можно увидеть своими глазами и пощупать руками (точнее, мышкой). Мы развеем миф о том, что «это слишком сложно».

---

### **А зачем мне лично это учить? Какая от этого польза?**

Давайте посмотрим на практические, осозаемые **выгоды**, которые вы получите, пройдя этот путь:

#### **● Вы обретете чувство уверенности и независимости.**

Вы больше не будете зависеть от того, приедет ли внук, чтобы «сделать красивенько» вашу страничку. Вы сможете изменить цвет, шрифт, расположение фотографий **сами**, когда захотите. Это чувство свободы и контроля бесценно.

#### **● Вы откроете в себе новые творческие способности.**

CSS — это **цифровое рукоделие**. Комбинации цветов, гармония отступов, игра со шрифтами — это эстетическое удовольствие, сравнимое с вышивкой или составлением букета. Вы сможете выражать себя через дизайн.

#### **● Вы создадите что-то настоящее и ценное.**

Результатом вашего обучения станет не просто «знание в голове», а реальный, живой продукт: ваша личная веб-страница, онлайн-фотоальбом, блог о путешествиях или сайт для вашего дачного кооператива. То, что можно показать друзьям, и чем можно гордиться.

#### **● Вы получите ключ к пониманию огромной части интернета.**

После этой книги вы будете смотреть на веб-страницы не как пользователь, а как **знаток**. Вы будете замечать: «О, здесь тень добавлена, тут интересное выравнивание, а эта кнопка меняет цвет при наведении — как мило!». Интернет откроется для вас с новой, технически-творческой стороны.

## **Это просто интересно и увлекательно!**

Процесс, когда из нескольких строчек простого кода на пустом экране рождается красота, затягивает. Это магия, секретами которой вы овладеете. Это хобби, которое не требует особых вложений, но дает огромную отдачу.

---

## **А что, если у меня ничего не получится?**

Получится обязательно. Потому что эта книга построена на трех китах:

- 1.Никакой спешки.** Мы будем возвращаться к сложным моментам столько раз, сколько нужно.
- 2.Огромное количество наглядных примеров.** Не просто «теория о рамках», а «давайте сделаем рамку для вашей любимой фотографии котика прямо сейчас».
- 3.Практика, практика и еще раз практика.** После каждой главы — задание, которое сразу дает видимый, радующий глаз результат.

**Эта книга — не учебник. Это приглашение.**

Приглашение в мир, где вы снова можете быть начинающими, но мудрыми и любопытными учениками. Где можно играть с цветами и формами, создавать красоту и удивлять самих себя.

Если вы хоть раз думали: «А неплохо бы сделать свою страничку в интернете... но это, наверное, очень сложно» — эта книга для вас. Она докажет, что это **не сложно**. Это **захватывающее**.

Готовы раскрасить свой цифровой мир? Тогда переворачивайте страницу. Начинаем наше путешествие.

- **Почему CSS – это не страшно, а очень увлекательно?**  
**(Сравнение с: ремонтом квартиры (HTML – стены, CSS – обои и расстановка мебели), вязанием, садоводством).**

Дорогой друг, представьте себе такую ситуацию: вам подарили чистый альбом и набор цветных карандашей. Страшно? Нет, конечно! Это же возможность рисовать что душе угодно. CSS — это и есть ваши волшебные карандаши для интернета. Давайте я докажу это через знакомые и приятные сравнения.

## 1. Ремонт в квартире — самое понятное сравнение

**HTML — это голая квартира после строителей.**

- Есть стены, окна, двери, пол и потолок. Всё на своих местах, всё функционально. Дверь открывается, окно закрывается. Но как-то пусто и неуютно.

**CSS — это ваш ремонт и дизайнерский подход.**

- `color: blue;` — поклеить голубые обои в спальне.
- `font-family: "Times New Roman";` — выбрать классические, строгие часы для гостиной.
- `margin: 20px;` — отодвинуть диван от стены, чтобы было удобнее ходить.
- `background-image: url("обои-с-розами.jpg");` — повесить любимую картину или фотообои.

**Что в этом увлекательного?** Вы не строите дом с нуля (это уже сделал HTML). Вы **обустраиваете** его, делаете уютным, индивидуальным, «своим». И если вам надоели голубые обои, вы не перестраиваете стену — вы просто меняете одну строчку кода (`color: pink;`) и комната сразу преображается. Это быстрее и чище, чем настоящий ремонт!

---

## 2. Вязание или вышивание — для творческих натур

Представьте, что **HTML — это основа, канва или схема вязания.**

- Это просто контуры, петли, обозначения. Безлико и технично.

### **CSS — это выбор ниток, цвета, фактуры и ваше мастерство.**

- `font-size: 24px;` — взять спицы потолще для крупной, выразительной вязки.
- `text-decoration: underline;` — вышить изящную кайму по краю.
- `border: 2px dotted red;` — связать ажурный край с красными узелками.
- `line-height: 1.6;` — сделать расстояние между рядами свободнее, чтобы узор «дышал».

**Что в этом увлекательного?** Это **медитативный, творческий процесс**. Вы видите, как под вашими руками (вернее, под внимательным взглядом и набором команд) рождается красота. Одна неправильная петля (ошибка в коде) — и вы её легко распускаете (исправляете) и вяжете заново. Результат — уникальное, созданное вами изделие (веб-страница), которого больше ни у кого нет.

---

## **3. Садоводство — для любителей живого и растущего**

### **HTML — это купленный дачный участок с намеченными грядками.**

- Вот здесь будет клумба, тут — огуречная грядка, там — яблоня. План есть, но земли не видно под схемой.

### **CSS — это ваша любовь, труд и искусство садовода.**

- `color: #228B22;` — полить молодую траву, чтобы она стала сочно-изумрудной.
- `padding: 15px;` — окучить кустик клубники, сделать для него уютное, защищённое пространство.
- `text-align: center;` — посадить розы в центре клумбы, сделав их главным акцентом.
- `box-shadow: 5px 5px 10px grey;` — поставить изящную садовую фигурку так, чтобы от неё падала лёгкая тень на дорожку.

**Что в этом увлекательного?** Вы **взаимодействуете с живым материалом** (ваших страницей). Вы ухаживаете за ней, формируете её облик. Сегодня добавили ярких красок (цветы-заголовки), завтра расчистили пространство (отступы). А «медиа-запросы» (о которых мы позже узнаем) — это как умная система полива, которая сама понимает, жарко сейчас на улице или прохладно (открыта страница на телефоне или на большом экране).

---

## Так почему же это **НЕ** страшно?

1. **Не нужно ничего изобретать с нуля.** Все «инструменты» (свойства CSS) уже придуманы. Ваша задача — научиться ими грамотно пользоваться, как художник учится смешивать краски из готовых тюбиков.
2. **Ошибки — это не катастрофа, а способ обучения.** Забыли точку с запятой? Страница «сломалась»? Ничего страшного! Вы не затопите соседей, как при ремонте. Просто посмотрите на код, найдите опечатку, исправьте — и всё снова заработает. Это безопасные, учебные ошибки.
3. **Мгновенный результат.** Это самое волшебное. Вы написали строчку кода, нажали «обновить» в браузере — и **ваша страница тут же изменилась на глазах**. Нет необходимости ждать, пока высохнет краска или вырастут цветы. Магия, которую вы творите своими руками, видна сразу.
4. **Можно всегда «откатиться».** Не понравился новый цвет фона? Верните старую строчку кода. Всё вернётся как было. Вы — полный хозяин положения.

**CSS — это язык визуальных возможностей.** Это не математика с единственным верным ответом. Это искусство, где есть миллионы способов сделать красиво, и ваш способ будет самым правильным, потому что он — **ваш**.

**И самое главное:** вам не нужно быть гением или технарем от рождения. Нужно лишь **любопытство** («а что будет, если я поменяю это число?») и **желание творить**. Всё остальное — дело практики, которую мы будем осваивать маленькими, радостными шагами.

**Готовы взять в руки свои цифровые кисти, спицы и садовые инструменты?** Тогда вперёд — к первой главе, где мы начнём с самого приятного: с выбора красок!

- **Что нам понадобится для начала? (Только компьютер, браузер и простейший блокнот (но... лучше — редактор кода Visual Studio Code!). Никаких сложных программ!).**

Дорогой друг, одно из главных преимуществ нашего занятия — для него не нужны дорогие инструменты или специальная мастерская. Всё, что необходимо, у вас уже, скорее всего, есть или легко и бесплатно появится. Давайте подготовим наш «рабочий уголок».

## 1. Основное оборудование (это есть у всех!)

- **Компьютер или ноутбук.** Подойдёт абсолютно любой, даже не самый новый. Не важно, Windows, macOS (MacBook) или Linux. Если он может открывать интернет — он нам подходит.
  - **Мышь.** Хотя многие операции можно делать с клавиатуры, с мышкой на первых порах будет гораздо удобнее и привычнее.
  - **Уверенность в себе.** Это — самый важный «инструмент». Помните: мы здесь не для экзамена, а для творчества и удовольствия.
- 

## 2. Программы-помощники (все бесплатные!)

### А. Браузер — наш «волшебный экран»

Это программа, в которой вы сейчас читаете этот текст (Google Chrome, Яндекс.Браузер, Mozilla Firefox, Safari).

- **Зачем?** В нём мы будем смотреть, как наши стили превращают скучную страницу в красивую. Это наш **главный холст**.
- **Рекомендация:** Установите **Google Chrome** или **Яндекс.Браузер**. Они очень дружелюбны к начинающим. Но если вы привыкли к другому браузеру — прекрасно, подойдёт любой.

## Б. Редактор кода — наш «умный блокнот»

Вот здесь — небольшой, но важный выбор. Мы можем пойти двумя путями:

### ● Путь «Максимально просто» (но неудобно):

- **Блокнот (Windows) илиTextEdit (Mac)** — они уже есть в вашем компьютере.
- **Минусы:** Они не «понимают» код. Не подсвечивают ошибки, не подсказывают команды. Как писать чернильной ручкой без линейки.

### ● Путь «Удобно и правильно» (рекомендуется!):

- **Visual Studio Code (сокращённо VS Code)** — это **бесплатный, современный и очень добрый к новичкам редактор**.
- **Почему он лучше блокнота?** Представьте разницу между:
  - **Писать от руки** (Блокнот).
  - **Писать в современном текстовом редакторе, который сам исправляет опечатки, подсказывает продолжение слов и красиво оформляет текст** (VS Code).

### ● Его волшебные способности:

- **Подсветка синтаксиса:** Команды CSS будут окрашиваться в свои цвета. `color` — в один цвет, `blue` — в другой. Ошибку увидите сразу!
- **Подсказки (IntelliSense):** Начнёте вводить `col...` — он предложит `color`, `column` и другие варианты. Не нужно всё помнить наизусть!
- **Автозакрытие:** Поставите открывающую фигурную скобку `{` — он сам добавит закрывающую `}`. Не потеряешься!
- **Проверка ошибок:** Если забыли точку с запятой, он мягко подчеркнёт это место цветом.

### ● Как его получить? Это проще, чем установить игру на телефон:

- Откройте браузер.
  - Введите в поиск: **«Скачать Visual Studio Code»**.
  - Зайдите на официальный сайт ([code.visualstudio.com](http://code.visualstudio.com)).
  - Нажмите большую синюю кнопку **«Download»**.
  - Запустите скачанный файл и нажимайте «Далее», не меняя настроек (можно просто соглашаться со всеми предложениями).
  - Готово! На рабочем столе появится ярлык с сине-фиолетовой иконкой.
- Не бойтесь этого слова «редактор кода».** Вы не будете там программировать ракеты. Вы будете просто писать текстовые

инструкции в очень удобной программе. Это как Microsoft Word, только для кода.

## Практическое задание «Нулевой уровень»:

1. **Проверьте:** У вас есть компьютер/ноутбук и браузер. 
  2. **Скачайте и установите** Visual Studio Code. Не торопитесь. Если возникнут сложности — это нормально! Попросите помощи у знакомых, детей или внуков. Один раз установить — и забыть об этом шаге навсегда.
  3. **Создайте на Рабочем столе папку** с понятным названием, например, **«Мой\_Веб\_Сайт»**.
    - Это будет наша «рабочая тетрадь», наш «проект». Все файлы будем хранить там, чтобы не потеряться.
- 

## 3. Что у нас будет в этой папке? (Структура проекта)

Сначала это будет выглядеть очень просто, как рецепт из двух ингредиентов:

text

Мой\_Веб\_Сайт (папка)

```
├── index.html    (Это файл с текстом и структурой страницы)  
└── style.css     (Это файл со всеми нашими стилями и красками!)
```

- Файл **.html** — это «скелет», содержание.
- Файл **.css** — это «одежда», дизайн.

Мы будем создавать эти файлы прямо в папке и открывать их в VS Code для редактирования, а потом смотреть результат в браузере.

---

## 4. Ваша главная «суперсила» — кнопка F12 (Инструменты разработчика)

В любом браузере есть **волшебная клавиша F12** (или комбинация Ctrl+Shift+I в Windows/Linux, Cmd+Option+I на Mac).

- **Что это?** Это секретное окошко, которое открывается прямо поверх вашей страницы.
  - **Зачем нам?** Это наш **микроскоп и обучающий тренажёр**. С его помощью можно:
    1. **Смотреть, как сделаны другие сайты.** Навели мышку на элемент — увидели, какие CSS-стили к нему применены. Это как заглянуть «под капот» любой красивой страницы!
    2. **Экспериментировать без риска.** В этом окошке можно менять стили **ЛЮБОЙ** страницы в интернете (цвет текста, размеры) — эти изменения будут видны только вам и исчезнут после обновления страницы.  
Идеальный полигон для проб!
  - **Мы будем активно им пользоваться.** Это снимет 90% страха «а вдруг я всё сломаю».
- 

## Итог нашей подготовки:

### Вам **НЕ** понадобится:

- Покупать лицензии на программы.
- Устанавливать сложные среды разработки.
- Иметь мощный игровой компьютер.
- Знать английский язык в совершенстве (команды мы выучим, они немногочисленны).

### Вам **ПОНАДОБИТСЯ** и **УЖЕ ЕСТЬ**:

1. Любой компьютер + браузер.
2. Бесплатный редактор **Visual Studio Code** (настоятельно рекомендуем!).
3. Папка для проекта на Рабочем столе.
4. Любопытство и готовность нажимать кнопки.

### Ваш рабочий процесс будет таким:

1. Пишем код в **VS Code** (удобно, с подсказками).
2. Сохраняем файл (**Ctrl+S**).
3. Переходим в браузер и **обновляем страницу** (**F5**).
4. **ВИДИМ РЕЗУЛЬТАТ** своих правок сразу.
5. Если что-то не так — исправляем в VS Code и снова обновляем.

Видите, как всё просто и логично? Мы создаём замкнутый, безопасный и мгновенно отзывающийся круг творчества.

**Когда ваш редактор VS Code будет установлен и папка «Мой\_Веб\_Сайт» создана — вы полностью готовы к старту. Мы начинаем наше путешествие с самой приятной части: с выбора красок! 🎨**

## ● **Важное правило: Не бояться экспериментировать и ошибаться! У нас всё безопасно.**

Дорогой друг, перед нами самое важное правило, которое определит весь успех нашего путешествия. Оно важнее всех технических знаний. Запомните его как мантру:

**«У нас всё безопасно. Здесь нельзя ничего сломать безвозвратно»**

Давайте разберём, почему это так, и как превратить страх в азарт первооткрывателя.

---

## **Почему мы боимся ошибиться с компьютером? (Разберём страхи по полочкам)**

### **1. «А вдруг я сломаю компьютер?» — НЕТ.**

- CSS — это не система управления компьютером. Это язык оформления. Максимум, что может случиться — ваша веб-страница в браузере будет выглядеть странно. Компьютер, операционная система, ваши фотографии и документы — всё это в полной безопасности. Это как если бы вы перепутали краски при рисовании картины — холст и кисти не пострадают.

### **2. «Я удалю что-то важное и не смогу вернуть» — НЕВОЗМОЖНО.**

- Мы работаем в текстовых файлах (.html и .css). Любой современный редактор (особенно VS Code) автоматически сохраняет историю. Вы можете нажать `Ctrl+Z` (отмена) десятки раз и вернуть всё как было. Это проще, чем распустить ряд в вязании!

### **3. «У меня ничего не получится, я не технарь» — ЗАБЛУЖДЕНИЕ.**

- CSS — это не высшая математика. Это логика и эстетика. Здесь нет единственного правильного ответа. Синий или зелёный? 10 пикселей или 15? Это вопросы вкуса, а не формулы. Ваш вариант — не ошибка, а **альтернативное решение**.
-

# Наша философия: Ошибка = Подсказка = Новое знание

Давайте переосмыслим само слово «ошибка». В нашем деле это не провал, а **самый эффективный учитель**.

## Как выглядит типичная «ошибка» в CSS?

- Вы написали `color: read;` вместо `color: red;`
- Вы забыли поставить `;` в конце строки.
- Вы использовали `margen` вместо `margin`.

**Что происходит?** Страница в браузере отображается не так, как вы ожидали. Может, текст не стал красным, или блок съехал не туда.

## И что же делать? Включаем режим детектива!

1. **Спокойно смотрим на экран.** Не «Ой, всё пропало!», а «Интересно, почему оно так сделало?».
2. **Проверяем код построчно.** Компьютер очень буквalen. Он сделал именно то, что вы ему сказали. Наша задача — найти, где приказ был искаён.
3. **Используем волшебную кнопку F12.** Нажимаем её, наводим на «сломанный» элемент и смотрим — а какие стили браузер к нему применил? Часто там будет зачёркнутая строка с жёлтым восклицательным знаком — это браузер прямо говорит: «Эту команду я не понял и проигнорировал».

**Вот момент магии:** Когда вы сами найдёте опечатку и исправите её, нажав `Ctrl+S` и обновив страницу (`F5`), — вы испытаете **триумф**. Небольшую, но очень важную победу над непониманием. И эта найденная ошибка навсегда врежется в память. Вы больше никогда не напишете `margen`.

---

## Лаборатория для смелых экспериментов

У нас есть два идеальных, безопасных полигона:

## 1. Файлы в вашей папке «Мой\_Веб\_Сайт»

Это ваша личная цифровая песочница. Делайте что хотите!

- **Хотите сделать фон ярко-розовым?** Пишите `background-color: hotpink;` и смотрите, что будет!
- **Интересно, что такое** `transform: rotate(45deg);`? Применяйте к картинке. Она повернётся на 45 градусов. Не понравилось — удалили строчку.
- **Что будет, если поставить шрифт размером в 200 пикселей?** Сделайте это! Увидите гигантскую букву. Смешно? Отлично! Теперь вы знаете, как это работает.

**Главный приём:** Комментируйте свои эксперименты.

В CSS можно писать «заметки на полях», которые компьютер игнорирует:

css

```
/* А это мой безумный эксперимент от 15 мая */  
.заголовок {  
    color: purple; /* Сделал фиолетовым */  
    /* border: 5px dotted green; - а этот стиль я пока выключил, посмотрю позже */  
}
```

Так вы не забудете, что и зачем делали.

## 2. Инструменты разработчика (F12) на ЛЮБОМ сайте

Это секретное оружие. Откройте любой любимый сайт (например, сайт о садоводстве или внучкин блог) и нажмите F12.

- **Найдите панель «Elements» (Элементы) и «Styles» (Стили).**
- **Щёлкните на значок лупы (выбор элемента) и наведите на любой блок на странице.**
- **Справа увидите все CSS-стили этого элемента. Попробуйте их менять прямо там!**
  - Щёлкните по цвету (`color`) и выберите другой из палитры.
  - Измените цифру у `font-size`. Текст на сайте тут же станет больше или меньше!
  - **ВСЁ, что вы меняете здесь, действует ТОЛЬКО У ВАС В БРАУЗЕРЕ.** Стоит обновить страницу — и сайт вернётся к первоначальному виду.

Это лучший способ учиться: разбирать, как сделаны профессиональные работы, и тут же пробовать свои идеи.

---

## Памятка бесстрашного веб-дизайнера

- Сохраняйтесь часто.** Привычка нажимать `Ctrl+S` после каждого изменения — ваш парашют.
- Действуйте методом «А что, если?»** Любая идея, даже самая безумная, достойна проверки. «А что, если заголовок будет зелёным, а не красным?» — Проверьте!
- Один шаг за раз.** Не пишите сразу 20 строк нового кода. Добавили одну строчку — сохранились — проверили в браузере. Так легко понять, какое именно изменение дало какой эффект (или ошибку).
- Хвалите себя за найденные ошибки.** Нашли пропущенную скобку? Вы — молодец! Это не промах, а успешно выполненная работа по отладке.
- Отдыхайте.** Если что-то упорно не получается, встаньте, попейте чаю, прогуляйтесь. Часто решение приходит само, когда мозг переключился.

**Запомните:** Самые красивые, нестандартные дизайнерские решения рождаются именно в процессе игры и экспериментов, а не при строгом следовании инструкциям.

**Ваша новая роль — не ученик, зазубривающий правила. Ваша роль — исследователь в собственной цифровой лаборатории. Вы смешиваете ингредиенты (свойства CSS) и наблюдаете за волшебной реакцией (отображением в браузере).**

Так давайте же начнём смешивать! Первый и самый приятный ингредиент — ЦВЕТ — ждёт нас во второй главе.

# Часть 1: Первые шаги. Знакомимся с кистями и красками

- Глава 1: Что такое CSS? Проще простого!
  - История в картинках: Страница без CSS (голые стены)  
→ Страница с CSS (уютный дом).

Дорогой друг, давайте совершим небольшое путешествие во времени. Мы наглядно увидим, как одна и та же информация может выглядеть совершенно по-разному. Это поможет вам понять самую суть того, ЧТО делает CSS.

## Сцена 1: Мир без CSS. «Голые стены»

Представьте себе новостройку. Вас пускают в квартиру после строителей. Вот что вы видите:

---

### Моё Хобби: Вязание

Я люблю вязать уже более 40 лет. Для меня это не просто рукоделие, а медитация, творчество и способ дарить близким тепло, сделанное своими руками.

#### Мои лучшие работы:

- Скатерть «Снежинка» (ажурная, круглая)
- Плед для внука «Звёздное небо» (тёплый, с узором)
- Набор ажурных салфеток для сервировки стола
- Шерстяные носки с оленями (для зимы)

#### Моё любимое фото работы:

[Здесь будет фото моей скатерти «Снежинка»]

#### Свяжитесь со мной, чтобы обсудить узоры!

Пишите на почту: [babushka@example.com](mailto:babushka@example.com)

С любовью, Галина Ивановна ❤

## Что мы видим в этом «цифровом помещении»?

- Все элементы есть: заголовок, текст, список, контакты, место для фото.
- Всё на своих местах, всё **функционально**. Информацию можно прочитать.
- Но как же это выглядит? **Скучно, безлико, невыразительно.**

- Всё написано одним стандартным шрифтом (обычно — Times New Roman).
- Нет цвета, кроме чёрного на белом.
- Нет структуры, всё идёт сплошным потоком.
- Фотография (если бы она была) была бы приклеена прямо к стене, без рамки.
- Это **чистая информация без души**. Как голая бетонная стена.

**Технически: это чистый HTML.** Он отвечает на вопросы: «**Что это?**» (заголовок, абзац, список) и «**В каком порядке?**».

---

## Сцена 2: Мы добавляем CSS. «Делаем ремонт»

Теперь мы становимся дизайнерами. Берём в руки наши инструменты (CSS) и начинаем преображение. Давайте посмотрим, как каждый шаг меняет облик нашей страницы.

### Шаг 1: Добавляем цвет и выбираем шрифт (клеим обои, вешаем люстру)

css

```
body {  
    font-family: 'Georgia', serif; /* Меняем стандартный шрифт на элегантный */  
    color: #333; /* Меняем угольно-чёрный на тёплый тёмно-серый */  
    background-color: #f9f5f0; /* Делаем фон не белым, а кремовым, как слоновая кость */  
}  
  
h1 {  
    color: #8b4513; /* Заголовок делаем цвета тёплого дерева (коричневый) */  
}
```

## Моё Хобби: Вязание

Я люблю вязать уже более 40 лет. Для меня это не просто рукоделие, а медитация, творчество и способ дарить близким тепло, сделанное своими руками.

### Мои лучшие работы:

- Скатерть «Снежинка» (ажурная, круглая)
- Плед для внука «Звёздное небо» (тёплый, с узором)
- Набор ажурных салфеток для сервировки стола
- Шерстяные носки с оленями (для зимы)

### Моё любимое фото работы:

[Здесь будет фото моей скатерти «Снежинка»]

### Свяжитесь со мной, чтобы обсудить узоры!

Пишите на почту: [babushka@example.com](mailto:babushka@example.com)

С любовью, Галина Ивановна ❤

**Что изменилось?** Появилась **атмосфера**. Страница стала теплее, уютнее, приобрела характер. Это уже не казённое помещение.

---

## Шаг 2: Расставляем мебель (работаем с отступами и блоками)

css

```
h1 {  
    text-align: center; /* Ставим заголовок по центру, как картину на стену */  
    margin-top: 40px; /* Опускаем его ниже от верха */  
    margin-bottom: 30px; /* И добавляем воздух снизу */  
}  
  
p {  
    line-height: 1.6; /* Увеличиваем расстояние между строками, чтобы текст "дышил" */  
    margin-left: 20px; /* Делаем небольшой отступ слева */  
}
```

## Моё Хобби: Вязание

Я люблю вязать уже более 40 лет. Для меня это не просто рукоделие, а медитация, творчество и способ дарить близким тепло, сделанное своими руками.

### Мои лучшие работы:

- Скатерть «Снежинка» (ажурная, круглая)
- Плед для внука «Звёздное небо» (тёплый, с узором)
- Набор ажурных салфеток для сервировки стола
- Шерстяные носки с оленями (для зимы)

### Моё любимое фото работы:

[Здесь будет фото моей скатерти «Снежинка»]

### Свяжитесь со мной, чтобы обсудить узоры!

Пишите на почту: [babushka@example.com](mailto:babushka@example.com)

С любовью, Галина Ивановна ❤

**Что изменилось?** Появился **порядок и воздух**. Элементы перестали давить друг на друга. Страница стала структурированной и удобной для глаз.

---

## Шаг 3: Украшаем детали (рамки, тени, акценты)

css

```
ul {  
    background-color: #ffffaf5; /* Подложка для списка - светло-бежевая */  
    padding: 20px;           /* Внутренние отступы - "воздух" внутри блока */  
    border-left: 5px solid #d2b48c; /* Тонкая коричневая рамка слева */  
    border-radius: 5px;       /* Слегка скругляем углы блока */  
    box-shadow: 2px 2px 5px rgba(0,0,0,0.05); /* Лёгкая, едва заметная тень */  
}  
  
a {  
    color: #a0522d; /* Цвет ссылок - терракотовый */  
    text-decoration: none; /* Убираем стандартное подчёркивание */  
}  
  
a:hover {  
    text-decoration: underline; /* Но добавляем его, когда наводим мышку */  
}
```

**Моё Хобби: Вязание**

Я люблю вязать уже более 40 лет. Для меня это не просто рукоделие, а медитация, творчество и способ дарить близким тепло, сделанное своими руками.

**Мои лучшие работы:**

- Скатерть «Снежинка» (ажурная, круглая)
- Плед для внука «Звёздное небо» (тёплый, с узором)
- Набор ажурных салфеток для сервировки стола
- Шерстяные носки с оленями (для зимы)

**Моё любимое фото работы:**

[Здесь будет фото моей скатерти «Снежинка»]

**Свяжитесь со мной, чтобы обсудить узоры!**

Пишите на почту: [babushka@example.com](mailto:babushka@example.com)

## ФИНАЛЬНАЯ СЦЕНА: Уютный дом, наполненный вашей личностью

Вот что мы в итоге получаем:

## ДО (Без CSS):

Голая информационная комната. Холодная, безликая, функциональная, но не привлекательная. Как склад.

## ПОСЛЕ (С CSS):

Ваш личный, тёплый, уютный уголок в интернете.

- Здесь есть **ваша цветовая гамма** (тёплые, пастельные тона).
- Здесь есть **логичная расстановка** (всё на своих местах, есть "воздух").
- Здесь есть **стильные детали** (аккуратные рамки, мягкие тени).
- Здесь есть **интерактивность** (ссылки реагируют на наведение).
- Здесь чувствуется **ваш характер, ваше хобби, ваша душа**.

## Суть метафоры «Ремонт»

- **HTML (стены, окна, двери):** Заголовок `<h1>`, абзац `<p>`, список `<ul>`, ссылка `<a>`, изображение `<img>`. Это **несущая конструкция**.
- **CSS (обои, краска, мебель, декор):** `color`, `font-family`, `margin`, `background-color`, `border`, `box-shadow`. Это **оформление и уют**.

**Самое главное:** Вы можете кардинально изменить дизайн (сделать «евроремонт» или «деревенский стиль»), **не трогая сам контент (HTML)**. Хотите вместо тёплых тонов — холодные синие? Поменяли 5 значений цвета в CSS — и страница преобразилась. Текст, фотографии, структура остались прежними!

**Вот почему CSS — это волшебство.** Вы берёте скелет информации и вдыхаете в него жизнь, настроение, индивидуальность. И делаете это с помощью простых, понятных команд, которые мы будем изучать шаг за шагом.

## Полный пошаговый код для нашего примера «Моё хобби: Вязание»

Дорогой друг, ниже вы найдёте два файла. Скопируйте их **дословно** в две разные программы (блокнот или VS Code) и сохраните в одну папку. После этого откройте файл `index.html` в браузере, и вы увидите всю магию превращения.

## ШАГ 0: Создайте папку

На Рабочем столе создайте новую папку с названием **МоеХобби**.

---

## ШАГ 1: Файл index.html (Каркас — «голые стены»)

Откройте **Блокнот** (Windows) или **VS Code**. Скопируйте этот код:

```
html
<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Моё хобби: Вязание</title>
    <!-- ВАЖНАЯ СТРОЧКА-СЕКРЕТ: Подключаем файл со стилями -->
    <link rel="stylesheet" href="style.css">
</head>
<body>

    <h1>Моё Хобби: Вязание</h1>

    <p>Я люблю вязать уже более 40 лет. Для меня это не просто рукоделие, а медитация, творчество и способ дарить близким тепло, сделанное своими руками.</p>

    <h2>Мои лучшие работы:</h2>
    <ul>
        <li>Скатерть «Снежинка» (ажурная, круглая)</li>
        <li>Плед для внука «Звёздное небо» (тёплый, с узором)</li>
        <li>Набор ажурных салфеток для сервировки стола</li>
        <li>Шерстяные носки с оленями (для зимы)</li>
    </ul>

    <h2>Моё любимое фото работы:</h2>
    <!-- Пока вместо фото просто рамка-заглушка. Позже можно вставить реальное фото! -->
    <div class="фото-рамка">
        [Здесь будет фото моей скатерти «Снежинка»]
    </div>
```

```
<h2>Свяжитесь со мной, чтобы обсудить узоры!</h2>
<p>Пишите на почту: <a href="mailto:babushka@example.com">babushka@example.com</a></p>

<p class="подпись">С любовью, Галина Ивановна ❤</p>

</body>
</html>
```

### Как сохранить:

1. В меню выберите **Файл → Сохранить как...**
  2. В поле «Тип файла» выберите **«Все файлы»**.
  3. В поле «Имя файла» напишите: index.html (обязательно с .html в конце!).
  4. Сохраните файл в папку **МоеХобби**.
- 

## ШАГ 2: Файл style.css (Дизайн — «ремонт и уют»)

Теперь создайте **НОВЫЙ файл** в Блокноте или VS Code. Скопируйте этот код:

css

```
/*
=====
ФАЙЛ СТИЛЕЙ ДЛЯ СТРАНИЦЫ О ХОББИ
(Можно менять всё, что угодно!)
===== */

/* 1. ОСНОВНОЙ ФОН И ШРИФТ (Делаем "стены" и "воздух") */
body {
    font-family: 'Georgia', 'Times New Roman', serif; /* Красивый, классический шрифт */
    font-size: 18px; /* Чуть увеличиваем базовый размер букв */
    line-height: 1.6; /* Делаем расстояние между строками побольше - легче читать! */
    color: #333333; /* Цвет текста - не чёрный, а тёмно-серый (мягче) */
    background-color: #f9f5f0; /* Цвет фона всей страницы - кремовый, как слоновая кость */
    margin: 0; /* Убираем стандартные отступы по краям окна */
    padding: 20px; /* Добавляем "воздух" от краёв окна до текста */
    max-width: 800px; /* Ограничиваем ширину текста, чтобы строки не были слишком длинными
*/
    margin-left: auto; /* Эти две строчки центрируют блок */
    margin-right: auto;
}
```

```
/* 2. ЗАГОЛОВОК ПЕРВОГО УРОВНЯ (h1) */
h1 {
    color: #8b4513; /* Цвет "коричневое седло" - тёплый, древесный */
    text-align: center; /* Выравниваем по центру */
    border-bottom: 2px solid #d2b48c; /* Тонкая линия под заголовком, цвет "загар" */
    padding-bottom: 15px; /* Отступ между текстом и линией */
    margin-top: 30px; /* Отступ сверху */
    margin-bottom: 40px; /* Большой отступ снизу */
}

/* 3. ЗАГОЛОВКИ ВТОРОГО УРОВНЯ (h2) */
h2 {
    color: #a0522d; /* Цвет "сцена" - ещё один тёплый коричневый */
    margin-top: 35px; /* Отступ сверху перед заголовком */
    padding-left: 10px; /* Небольшой отступ слева */
    border-left: 4px solid #d2b48c; /* Цветная полоска слева от заголовка */
}

/* 4. ОБЫЧНЫЕ АБЗАЦЫ (p) */
p {
    margin-bottom: 20px; /* Отступ снизу у каждого абзаца */
    text-align: justify; /* Выравнивание по ширине, ровные края */
}

/* 5. СПИСОК (ul и li) - Делаем красивую "карточку" */
ul {
    background-color: #ffffff; /* Белый фон для списка */
    padding: 25px 25px 25px 40px; /* Внутренние отступы: сверху, справа, снизу, слева */
    border-radius: 10px; /* Скругляем углы белого блока */
    border-left: 8px solid #d2b48c; /* Толстая цветная полоса слева */
    box-shadow: 3px 3px 10px rgba(0, 0, 0, 0.08); /* ЛЁГКАЯ тень вокруг блока */
    /* Тень: смещение по X, по Y, размытие, цвет (чёрный с прозрачностью 8%) */
}

li {
    margin-bottom: 10px; /* Отступ между пунктами списка */
    font-size: 17px; /* Размер шрифта в списке */
}

/* 6. РАМКА ДЛЯ ФОТОГРАФИИ (пока это просто блок-заглушка) */
.фото-рамка {
    background-color: #ffffaf; /* Очень светлый бежевый фон */
    border: 3px solid #e6d5c3; /* Светлая рамка */
    border-radius: 5px; /* Слегка скруглённые углы у рамки */
```

```

padding: 40px; /* Большой внутренний отступ - "поле" внутри рамки */
text-align: center; /* Текст по центру */
color: #b8860b; /* Цвет текста внутри рамки - золотистый */
font-style: italic; /* Курсив */
margin: 25px 0; /* Отступы сверху и снизу */
box-shadow: 2px 2px 8px rgba(0, 0, 0, 0.05); /* Ещё более лёгкая тень */
}

/* 7. ССЫЛКИ (a) */
a {
    color: #a0522d; /* Терракотовый цвет для ссылок */
    text-decoration: none; /* Убираем стандартное подчёркивание */
    font-weight: bold; /* Делаем жирными */
}

a:hover {
    text-decoration: underline; /* Подчёркиваем ссылку ТОЛЬКО когда наводим курсор мыши */
    color: #8b0000; /* Меняем цвет на тёмно-красный при наведении */
}

/* 8. ОСОБАЯ ПОДПИСЬ В КОНЦЕ */
.подпись {
    text-align: right; /* Выравниваем по правому краю */
    font-size: 20px; /* Чуть крупнее */
    color: #8b4513; /* Коричневый цвет */
    font-style: italic; /* Курсив */
    margin-top: 50px; /* Большой отступ сверху */
    padding-right: 20px; /* Отступ справа */
}

```

### Как сохранить:

1. **Файл → Сохранить как...**
  2. **Тип файла:** «Все файлы»
  3. **Имя файла:** style.css (обязательно .css!)
  4. Сохраните в **ТУ ЖЕ САМУЮ папку** МоеХобби, где лежит index.html.
- 

## ШАГ 3: Волшебное действие

Теперь найдите в папке МоеХобби файл index.html. Он, скорее всего, будет отображаться как значок вашего браузера.

**Просто ДВАЖДЫ ЩЁЛКНИТЕ по нему левой кнопкой мыши.** Файл автоматически откроется в вашем браузере.

**Моё Хобби: Вязание**

Я люблю вязать уже более 40 лет. Для меня это не просто рукоделие, а медитация, творчество и способ дарить близким тепло, сделанное своими руками.

**Мои лучшие работы:**

- Скатерть «Снегинка» (ажурная, круглая)
- Плед для внука «Звёздное небо» (тёплый, с узором)
- Набор ажурных салфеток для сервировки стола
- Шерстяные носки с оленями (для зимы)

**Моё любимое фото работы:**

[Здесь будет фото моей скатерти «Снегинка»]

**Свяжитесь со мной, чтобы обсудить узоры!**

Пишите на почту: [babushka@example.com](mailto:babushka@example.com)

С любовью, Галина Ивановна ❤️

## Что вы увидите и как экспериментировать:

Вы увидите готовую, красивую страницу. Теперь самое интересное!

1. Откройте файл `style.css` в Блокноте/VSCode.
2. Найдите, например, строку `background-color: #f9f5f0;` в самом начале.
3. Измените цвет на любой другой:
  - `background-color: #e6f7ff;` (светло-голубой)
  - `background-color: #f0ffe6;` (светло-зелёный)
  - `background-color: #fff0f5;` (лавандовый, очень светлый розовый)
4. Сохраните файл (`Ctrl+S`).
5. Вернитесь в браузер и НАЖМИТЕ КНОПКУ `F5` (или иконку «Обновить»).

Страница МГНОВЕННО изменит цвет фона! Вы только что совершили своё первое волшебство с CSS.

# Важная техническая магия: Как файлы находят друг друга

Обратите внимание на **ВОЛШЕБНУЮ СТРОЧКУ** внутри файла index.html:

html

```
<link rel="stylesheet" href="style.css">
```

Эта строчка говорит браузеру: «Дорогой браузер, когда ты будешь показывать эту страницу, пожалуйста, найди в той же папке файл с именем style.css и примени все стили из него к моим элементам».

Главное правило: **оба файла (.html и .css) ДОЛЖНЫ лежать в ОДНОЙ ПАПКЕ**. Имя папки не имеет значения, главное — чтобы они были вместе.

Теперь у вас есть полный, работающий пример, с которым можно безопасно играть, меняя цифры и цвета!

## ■ Где живут стили? Три способа «прицепить» стиль к странице:

1. Рядом с элементом (как записка на холодильник) – inline-стили.
2. В «шапке» страницы (как инструкция для всей комнаты) – стили в `<head>`.
3. В отдельной «книге рецептов» (самый удобный способ!) – внешний файл `.css`.

Дорогой друг, представьте, что у вас есть инструкция по уходу за комнатным растением. Где вы её разместите? Можно написать на горшке, можно повесить на стену рядом, а можно завести отдельную тетрадь для всех растений. Так и со стилями CSS — их можно разместить в разных местах. Давайте разберём все три способа с живыми примерами.

### Способ 1. Рядом с элементом (как записка на холодильник) — Inline-стили

**Метафора:** Вы пишете записку «Не забыть купить молоко!» и приклеиваете её прямо на дверцу холодильника. Заметка прикреплена к одному конкретному предмету.

**Как это выглядит в коде:**

```
html  
<p style="color: red; font-size: 20px;">Этот текст будет красным и крупным.</p>  
<h1 style="background-color: yellow; text-align: center;">Важное объявление!</h1>
```

**Ключевые особенности:**

- Используется атрибут `style="..."` прямо внутри тега HTML.
- Стили действуют **только на этот один конкретный элемент**.
- Запись идёт **прямо в значениях**, без использования **селекторов** (о них позже).
- Между свойствами ставится точка с запятой `;`.

## **Преимущества:**

- **Очень быстро** — видим элемент, тут же его оформляем.
- **Высокий приоритет** — такой стиль «перекрывает» другие (если есть конфликт).

## **Недостатки:**

- **Непрактично** — если нужно изменить цвет 10 абзацев, придётся править 10 раз.
- **Загромождает** HTML-код, смешивая структуру и оформление.
- **Сложно править** — чтобы найти все красные заголовки, придётся просматривать всю страницу.

**Когда использовать:** Крайне редко. Только для **быстрых экспериментов** или уникального оформления **одного-единственного** элемента на странице, который точно больше нигде не повторится.

---

## **Способ 2. В «шапке» страницы (как инструкция для всей комнаты) — Стили в `<head>`**

**Метафора:** У вас есть инструкция «В этой комнате все картины вешаем на высоте 150 см от пола» и вы прикрепляете её на входную дверь. Правило действует на все объекты определённого типа в пределах одной комнаты.

## **Как это выглядит в коде:**

```
html
<!DOCTYPE html>
<html lang="ru">
<head>
  <meta charset="UTF-8">
  <title>Моя страница</title>

  <!-- А вот тут живут наши стили для всей страницы -->
  <style>
    /* Все абзацы на этой странице будут синими */
    p {
```

```

        color: blue;
        font-size: 18px;
        line-height: 1.5;
    }

/* Все заголовки первого уровня - зелёные и по центру */
h1 {
    color: green;
    text-align: center;
}

/* Все элементы с классом "важно" получат жёлтый фон */
.vажно {
    background-color: yellow;
    padding: 10px;
}

</style>

</head>
<body>
    <h1>Добро пожаловать!</h1>
    <p>Это абзац. Он автоматически будет синим.</p>
    <p>И этот тоже будет синим.</p>
    <div class="важно">А этот блок будет на жёлтом фоне.</div>
</body>
</html>

```

## Ключевые особенности:

- Стили помещаются между тегами `<style>` и `</style>`.
- Эти теги располагаются внутри раздела `<head>` (головы/шапки) документа.
- Здесь уже появляются **селекторы** (`p`, `h1`, `.важно`) — они указывают, К КАКИМ элементам применяются стили.
- Стили действуют на **всю текущую страницу**, но не на другие страницы сайта.

## Преимущества:

- **Удобно для одной страницы** — все стили собраны в одном месте, в начале документа.
- **Можно задавать правила для множества элементов** (все абзацы, все ссылки).
- **Чистый HTML** — структура страницы (тело) не загромождена стилями.

## Недостатки:

- Не подходит для сайта из нескольких страниц — если нужно одинаковое меню на 10 страницах, стили придётся копировать в каждую.
- Увеличивает размер HTML-файла.

**Когда использовать:** Для **небольших, одностораничных проектов** (например, персональная визитка, поздравительная открытка, рецепт). Или для быстрого прототипирования.

---

## Способ 3. В отдельной «книге рецептов» — Внешний файл .css

**Метафора:** У вас есть толстая тетрадь «Рецепты красивого дизайна». В ней на отдельных страницах: «Рецепт красивого заголовка», «Рецепт уютного абзаца», «Рецепт кнопки при наведении». Этой тетрадью можно пользоваться на любой кухне (на любой странице сайта).

**Как это выглядит:**

**Файл style.css (отдельный файл в той же папке):**

css

```
/* Это файл-библиотека стилей для всего моего сайта */
```

```
/* Рецепт для всех абзацев на сайте */
```

```
p {  
    font-family: 'Georgia', serif;  
    line-height: 1.6;  
    color: #333;  
    margin-bottom: 20px;  
}
```

```
/* Рецепт для главного заголовка */
```

```
h1 {  
    color: #8b4513;  
    text-align: center;  
    border-bottom: 2px solid #d2b48c;  
    padding-bottom: 15px;  
}
```

```
/* Рецепт для кнопок */
.кнопка {
    background-color: #4CAF50;
    color: white;
    padding: 12px 24px;
    border: none;
    border-radius: 5px;
    font-size: 16px;
}

/* Рецепт для кнопки ПРИ НАВЕДЕНИИ мыши */
.кнопка:hover {
    background-color: #45a049;
}
```

### Файл index.html (главная страница):

```
html

<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="UTF-8">
    <title>Мой сайт</title>
    <!-- ВАЖНО: Подключаем нашу "книгу рецептов"! -->
    <link rel="stylesheet" href="style.css">
</head>
<body>
    <h1>Мой прекрасный сайт</h1>
    <p>Этот абзац автоматически получит все стили из файла style.css</p>
    <button class="кнопка">Нажми меня</button>
</body>
</html>
```

### Файл about.html (другая страница того же сайта):

```
html

<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="UTF-8">
    <title>Обо мне</title>
    <!-- Та же самая "книга рецептов"! -->
    <link rel="stylesheet" href="style.css">
```

```
</head>
<body>
    <h1>Обо мне</h1>
    <p>И этот абзац будет оформлен так же, как на главной странице!</p>
    <!-- Та же кнопка с теми же эффектами -->
    <button class="кнопка">Написать мне</button>
</body>
</html>
```

### Ключевые особенности:

- Стили хранятся в **отдельном файле с расширением .css**.
- В HTML-файле в разделе `<head>` добавляется **тег-ссылка** `<link>`, который «подключает» этот CSS-файл.
- **Один CSS-файл можно подключить к ЛЮБОМУ количеству HTML-страниц.**
- Изменения, внесённые в CSS-файл, **автоматически применяются ко всем страницам**, которые его используют.

### Преимущества:

- **Единый стиль для всего сайта** — изменили цвет в одном месте, и он обновился на 100 страницах.
- **Чистота и порядок** — HTML отвечает за структуру, CSS — за оформление.
- **Кэширование браузером** — страницы загружаются быстрее.
- **Лёгкость поддержки** — все стили в одном месте.

### Недостатки:

- Нужно создавать и следить за **дополнительным файлом**.
- Надо **правильно подключать** файл (самая частая ошибка новичков — неправильный путь в `href`).

**Когда использовать: ВСЕГДА, КОГДА ДЕЛАЕТЕ САЙТ!** Это профессиональный, правильный и самый удобный подход.

---

## Сравнительная таблица «Что, когда и зачем»

Способ	Где пишем?	Для чего?	Приоритет	Пример из жизни
<b>Inline (в строке)</b>	Прямо в теге: <code>&lt;p style="...&gt;</code>	Быстрое точечное изменение одного элемента	<b>Самый высокий</b> (побеждает всех)	Записка на холодильнике
<b>В <code>&lt;head&gt;</code></b>	Между <code>&lt;style&gt;</code> тегами вверху страницы	Оформление одной конкретной страницы	Средний	Инструкция на двери комнаты
<b>Внешний файл</b>	В отдельном файле <code>.css</code> , подключаем через <code>&lt;link&gt;</code>	Единый стиль для всего сайта, профессиональная работа	Самый низкий (но этого нам и нужно!)	Книга рецептов на кухне

## Важное правило каскада (очень просто!)

Если к одному элементу применены стили разными способами, браузер решает, какой из них важнее, по простому принципу:

**«Ближайший к элементу стиль — самый важный».**

1. **Внешний файл (`.css`)** — дальше всех от элемента → самый низкий приоритет.
2. **Стили в `<head>`** — ближе → приоритет выше.
3. **Inline-стиль** — прямо на элементе → **побеждает всегда!**

**Практический совет:** Начинайте **сразу с 3-го способа** (отдельный файл `.css`). Так вы с самого начала учитесь делать правильно. Способы 1 и 2 используйте только для очень быстрых проверок и экспериментов.

**Попробуйте сейчас:** Откройте пример из предыдущей главы и попробуйте добавить к какому-нибудь элементу inline-стиль. Например:

html

< p style="color: purple; border: 2px solid purple;">Этот абзац будет фиолетовым с рамкой, несмотря на все другие стили!</p>

Убедитесь, что этот стиль «перебил» все остальные. Вот как работает приоритет!

Теперь вы знаете все три «адреса», по которым живут стили. Давайте перейдём к самому интересному — к тому, **какими красками** мы будем рисовать!

## ● Глава 2: Наш первый волшебный ингредиент – ЦВЕТ.

- Как говорить с компьютером о цвете? Простые слова (`red`, `blue`) и «волшебные коды» (#FF0000, `rgb(255,0,0)`).

Дорогой друг, представьте, что вы художник и просите помощника принести вам краску. Вы можете сказать: «Принеси красную» — и он принесёт какой-то красный оттенок. А можете сказать: «Принеси краску марки "Кадмий красный светлый", номер 123 в каталоге» — и получите именно тот оттенок, который вам нужен. Компьютер понимает оба языка! Давайте научимся говорить с ним на языке цвета.

### 1. Простые слова (ключевые слова) — как в детской раскраске

Самый понятный способ — использовать **английские названия цветов**. Их около 140, но нам хватит и двух десятков основных.

#### Базовые цвета (как в коробке с 12 карандашами):

css

```
p {  
    color: red; /* Красный */  
    color: blue; /* Синий */  
    color: green; /* Зелёный */  
    color: yellow; /* Жёлтый */  
    color: black; /* Чёрный */  
    color: white; /* Белый */  
    color: gray; /* Серый */  
    color: brown; /* Коричневый */  
    color: orange; /* Оранжевый */  
    color: purple; /* Фиолетовый */  
    color: pink; /* Розовый */  
}
```

#### Более интересные названия (как в большой коробке с 48 карандашами):

css

```
h1 {
```

```
color: crimson;      • Малиновый */
color: navy;        • Тёмно-синий (морской) */
color: teal;         • Бирюзовый */
color: olive;        • Оливковый */
color: coral;        • Коралловый */
color: lavender;     • Лавандовый */
color: chocolate;    • Шоколадный */
color: goldenrod;    • Золотистый */
}
```

## Прозрачность и специальные:

css

```
div {
  color: transparent; /* Полностью прозрачный */
  color: currentColor; /* Возьми цвет у родителя */
}
```

## Преимущества простых названий:

- Очень легко запомнить — red это красный, blue — синий
- Не нужно ничего вычислять — просто пишем слово
- Идеально для быстрых экспериментов

## Недостатки:

- Ограниченнная палитра — всего ~140 цветов
- Нет точных оттенков — какой именно синий вы хотите? Голубой? Тёмно-синий? Лазурный?
- Разные браузеры могут немного по-разному отображать cornflowerblue

**Когда использовать:** Для обучения, быстрых набросков, когда точный оттенок не важен.

---

## 2. «Волшебные коды» HEX — как номер краски в каталоге

Это самый популярный способ у веб-дизайнеров. Код всегда начинается с решётки # и содержит 6 символов (цифры от 0-9 и буквы A-F).

## Как устроен HEX-код: #RRGGBB

Представьте, что цвет состоит из трёх пробирок:

- **RR** — сколько **красного** (Red) от 00 (нет) до FF (максимум)
- **GG** — сколько **зелёного** (Green)
- **BB** — сколько **синего** (Blue)

**Примеры:**

css

```
/* Чистые цвета */  
color: #FF0000; /* Красный: MAX красного, НОЛЬ зелёного, НОЛЬ синего */  
color: #00FF00; /* Зелёный: ноль красного, MAX зелёного, ноль синего */  
color: #0000FF; /* Синий: ноль красного, ноль зелёного, MAX синего */  
  
/* Смешиваем: */  
color: #FFFF00; /* Жёлтый: красный + зелёный = #FF0000 + #00FF00 */  
color: #FF00FF; /* Пурпурный: красный + синий */  
color: #00FFFF; • Голубой (циан): зелёный + синий */  
  
/* Серые оттенки (когда все три компонента равны): */  
color: #000000; • Чёрный (все по нулям) */  
color: #808080; • Средне-серый */  
color: #C0C0C0; • Серебряный */  
color: #FFFFFF; • Белый (все по максимуму) */  
  
/* Более сложные, красивые цвета: */  
color: #8B4513; • Коричневый (седло) — наш любимый! */  
color: #FFD700; • Золотой */  
color: #FF69B4; • Ярко-розовый */  
color: #4B0082; • Индиго */
```

**Сокращённая запись (если все пары одинаковы):**

css

```
color: #F00; /* То же, что #FF0000 — красный */  
color: #0F0; • То же, что #00FF00 — зелёный */  
color: #00F; • То же, что #0000FF — синий */  
color: #CCC; • То же, что #CCCCCC — светло-серый */
```

**Как найти нужный HEX-код?**

1. **В редакторе VS Code** — начните вводить цвет, и появится палитра!
2. **В интернете** — поищите «HEX color picker» или «палитра цветов»
3. **В графических редакторах** (даже в Paint!) есть eyedropper (пипетка), которая показывает код цвета
4. **Специальные сайты** — [colorhunt.co](https://colorhunt.co), [coolors.co](https://coolors.co) — готовые красивые палитры

### Преимущества HEX:

- **Точность** — миллионы оттенков
  - **Стандарт** — все браузеры отображают одинаково
  - **Компактность** — короткая запись
  - **Удобно копировать** — скопировал код, вставил, получил тот же цвет
- 

## 3. RGB и RGBA — как смешивать краски на палитре

Ещё более понятный способ! Вы прямо говорите компьютеру: «Смешай 120 частей красного, 80 частей зелёного и 200 частей синего».

### RGB (Red, Green, Blue)

Каждый компонент от 0 до 255:

css

```
color: rgb(255, 0, 0); /* Красный */
color: rgb(0, 255, 0); • Зелёный */
color: rgb(0, 0, 255); • Синий */
color: rgb(255, 255, 0); • Жёлтый */
color: rgb(139, 69, 19); • Коричневый (тот же, что #8B4513) */
```

### RGBA — то же самое, но с прозрачностью!

Буква **A** — Alpha (прозрачность) от 0 (полностью прозрачный) до 1 (полностью непрозрачный):

css

```
background-color: rgba(255, 0, 0, 0.5); • Красный, полупрозрачный (50%) */
background-color: rgba(0, 0, 0, 0.3); • Чёрный, слегка прозрачный (30%) */
background-color: rgba(255, 255, 255, 0.8); • Белый, почти непрозрачный (80%) */
```

### Преимущества RGB/RGBA:

- **Очень наглядно** — сразу видно, сколько каждой краски
  - **Можно делать прозрачность (RGBA)**
  - **Легче подбирать похожие оттенки** — чуть увеличил красный, получил теплее
- 

## 4. HSL и HSLA — как художник-колорист

Самый творческий способ! Вы говорите не о количестве красок, а о:

- **Hue (Оттенок)** — от 0 до 360° (цветовой круг)
- **Saturation (Насыщенность)** — от 0% (серый) до 100% (яркий)
- **Lightness (Светлота)** — от 0% (чёрный) до 100% (белый)

css

```
/* Все чистые цвета на цветовом круге: */
color: hsl(0, 100%, 50%); • Красный */
color: hsl(120, 100%, 50%); • Зелёный */
color: hsl(240, 100%, 50%); • Синий */

/* Менее насыщенные: */
color: hsl(0, 50%, 50%); • Приглушенный красный */
color: hsl(120, 30%, 70%); • Светлый, пастельный зелёный */

/* Разные светлоты: */
color: hsl(0, 100%, 25%); • Тёмно-красный */
color: hsl(0, 100%, 75%); • Светло-красный */

/* Серые (насыщенность 0%): */
color: hsl(0, 0%, 50%); • Средне-серый */
```

### HSLA — с прозрачностью:

css

```
background-color: hsla(240, 100%, 50%, 0.3); /* Синий, прозрачный 30% */
```

### Преимущества HSL:

- Очень легко создавать гармоничные палитры — меняешь только тон (H), сохраняя насыщенность и светлоту
  - Интуитивно понятно — «хочу более светлый синий» = увеличиваешь Lightness
  - Проще подбирать сочетания цветов
- 

## Практикум: Давайте поиграем с цветами!

Создайте файл `index.html` и скопируйте этот код:

```
html
<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="UTF-8">
    <title>Играем с цветами</title>
    <style>
        body {
            font-family: Arial, sans-serif;
            padding: 30px;
            background-color: #f0f8ff;
        }

        .цветовой-блок {
            padding: 20px;
            margin: 15px 0;
            border-radius: 10px;
            color: white;
            font-weight: bold;
            box-shadow: 3px 3px 10px rgba(0,0,0,0.1);
        }
    </style>
</head>
<body>
    <h1>Волшебство цветов CSS</h1>

    <div class="цветовой-блок" style="background-color: red;">
        1. Простое слово: red
    </div>
```

```

<div class="цветовой-блок" style="background-color: #FF6347;">
    2. HEX-код: #FF6347 (томатный)
</div>

<div class="цветовой-блок" style="background-color: rgb(70, 130, 180);">
    3. RGB: rgb(70, 130, 180) (стальной синий)
</div>

<div class="цветовой-блок" style="background-color: rgba(255, 99, 71, 0.7);">
    4. RGBA с прозрачностью: rgba(255, 99, 71, 0.7)
</div>

<div class="цветовой-блок" style="background-color: hsl(39, 100%, 50%); color: black;">
    5. HSL: hsl(39, 100%, 50%) (оранжевый)
</div>

<div class="цветовой-блок" style="background-color: hsla(120, 100%, 25%, 0.5);">
    6. HSLA: hsla(120, 100%, 25%, 0.5) (полупрозрачный тёмно-зелёный)
</div>

<p>Попробуйте изменить значения и посмотреть, что получится!</p>

<h2>Мои эксперименты:</h2>
<div class="цветовой-блок" style="background-color: #8B4513;">
    Мой любимый цвет: #8B4513 (коричневый седло)
</div>

<div class="цветовой-блок" style="background-color: rgb(218, 165, 32);">
    Золотистый: rgb(218, 165, 32)
</div>
</body>
</html>

```

## Что делать:

- Сохраните этот файл как `index.html`
  - Откройте его в браузере — увидите все способы задания цвета
  - Изменяйте значения в коде, сохраняйте (`Ctrl+S`) и обновляйте страницу (`F5`)
  - Попробуйте:
    - Сделать фон у первого блока `dodgerblue` (простое слово)
    - Изменить `#FF6347` на `#6495ED` (HEX)
    - Сделать последний блок полупрозрачным, добавив `rgba()` или `hsla()`
-

# Шпаргалка «Какой способ выбрать?»

Задача	Лучший способ Пример	
Быстро, для обучения	Простые слова	color: blue;
Точный цвет, веб-стандарт	HEX-код	color: #8B4513;
Нужна прозрачность	RGBA или HSLA	background: rgba(0,0,0,0.5);
Создаю цветовую палитру	HSL	hsl(200, 70%, 50%) → hsl(200, 70%, 60%)
Цвет из графического редактора HEX (чаще всего)	Скопировали из Photoshop/Figma	

**Самый важный совет:** Не пытайтесь запомнить все коды! **Никто не помнит все HEX-коды наизусть.** Все дизайнеры:

1. Используют палитры в редакторах
2. Копируют коды с сайтов
3. Экспериментируют, меняя значения

## Ваше домашнее задание:

1. Создайте папку **МоиЦвета**
2. В ней создайте файл **index.html**
3. Сделайте 5 абзацев текста разными способами задания цвета
4. Сделайте фон страницы цветом **honeydew** (простое слово)
5. Сделайте рамку вокруг одного абзаца цветом **#D2691E** (HEX)

**Запомните:** Компьютер — ваш терпеливый помощник. Он будет пробовать любой цвет, который вы ему дадите. Даже если это будет **color: unicorn;** (нет, это не сработает, но попробовать можно!). Главное — начать играть с цветами, и вы быстро почувствуете себя настоящим цифровым художником! 

## ■ Практикум №1: Раскрашиваем заголовок и абзац. Видим результат мгновенно!

Дорогой друг, пришло время настоящей магии! Сейчас мы создадим простую страницу и будем менять её цвета прямо на ваших глазах. Вы увидите, как одно маленькое изменение в коде мгновенно преображает страницу в браузере.

### ШАГ 1: Создаём нашу «цифровую мастерскую»

#### 1.1. Создайте папку для практикума

На Рабочем столе создайте новую папку с понятным названием:

- Практикум1\_Цвета
- или МояПерваяРаскраска

#### 1.2. Создайте HTML-файл

Откройте **Блокнот** или **VS Code** и создайте новый файл. Скопируйте этот базовый код:

```
html

<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Моя первая раскраска</title>
    <!-- Здесь скоро подключим стили -->
</head>
<body>
    <h1>Добро пожаловать в мой сад!</h1>

    <p>Сегодня я хочу рассказать вам о моих любимых цветах, которые растут на даче. Каждое утро, выходя в сад, я любуюсь этой красотой.</p>
```

```
<p>Розы — мои самые любимые. У меня есть красные, розовые и белые сорта. Они пахнут так волшебно, что пчёлы прилетают даже с соседнего участка.</p>
```

```
<h2>Что нужно цветам для счастья:</h2>
```

```
<ul>
  <li>Солнечный свет</li>
  <li>Своевременный полив</li>
  <li>Любовь и забота</li>
</ul>
```

```
<p>Приезжайте в гости летом — покажу всё самое красивое!</p>
```

```
<p>С теплом, Ваша соседка-садовод 
```

```
</body>
</html>
```

### Как сохранить:

1. Нажмите Ctrl+S (Сохранить)
2. Выберите вашу папку Практикум1\_Цвета
3. В поле «Имя файла» напишите: index.html
4. В поле «Тип файла» выберите «**Все файлы**»
5. Нажмите «Сохранить»

### 1.3. Проверьте, что файл работает

Найдите в папке файл index.html и **дважды щёлкните** по нему левой кнопкой мыши. Он откроется в вашем браузере.

Вы увидите чёрный текст на белом фоне — это наша «неокрашенная» страница. Пока она выглядит скучно, но это ненадолго!

---

## ШАГ 2: Добавляем CSS прямо в HTML (самый быстрый способ)

Мы начнём со второго способа, который изучили — стили внутри `<head>`. Это позволит нам сразу видеть результат.

## 2.1. Добавьте стили в файл index.html

Вернитесь в редактор с файлом index.html. Найдите строку:

html

```
<!-- Здесь скоро подключим стили -->
```

И замените её на вот этот код:

html

```
<style>
/* СЕКРЕТНАЯ ЛАБОРАТОРИЯ ЦВЕТА */

/* 1. Меняем фон всей страницы */
body {
    background-color: #f0ffff; /* Очень светлый зелёный, как молодая трава */
    padding: 30px; /* Добавляем "воздух" по краям */
    font-family: 'Georgia', serif; /* Красивый классический шрифт */
}

/* 2. Раскрашиваем главный заголовок */
h1 {
    color: #2e8b57; /* Цвет морской волны */
    text-align: center; /* Размещаем по центру */
    border-bottom: 3px solid #98fb98; /* Зелёная линия под заголовком */
    padding-bottom: 15px; /* Отступ между текстом и линией */
}

/* 3. Раскрашиваем обычные абзацы */
p {
    color: #228b22; /* Лесной зелёный */
    font-size: 18px; /* Чуть увеличиваем буквы */
    line-height: 1.6; /* Делаем расстояние между строками побольше */
}

/* 4. Раскрашиваем подзаголовок */
h2 {
    color: #8b4513; /* Коричневый, как земля */
    background-color: #ffffacd; /* Светло-жёлтый фон */
    display: inline-block; /* Чтобы фон был только вокруг текста */
    padding: 8px 15px; /* Отступы внутри жёлтого блока */
    border-radius: 10px; /* Закругляем углы блока */
}
```

```

/* 5. Раскрашиваем список */
ul {
    color: #556b2f; /* Тёмный оливковый */
    background-color: #fafad2; /* Светло-золотистый фон */
    padding: 20px 20px 20px 40px; /* Отступы: сверху, справа, снизу, слева */
    border-radius: 15px; /* Сильно закругляем углы */
    border-left: 8px solid #daa520; /* Золотистая полоса слева */
}

li {
    margin-bottom: 8px; /* Отступ между пунктами списка */
    font-size: 17px;
}

/* 6. Особый стиль для последнего абзаца */
p:last-child {
    color: #b22222; /* Огненно-красный */
    font-style: italic; /* Курсив */
    text-align: right; /* Выравниваем по правому краю */
    font-size: 20px; /* Делаем чуть крупнее */
}

```

</style>

**Ваш полный файл index.html теперь должен выглядеть так:**

```

html
<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Моя первая раскраска</title>
    <style>
        /* СЕКРЕТНАЯ ЛАБОРАТОРИЯ ЦВЕТА */

        /* 1. Меняем фон всей страницы */
        body {
            background-color: #f0ffff; /* Очень светлый зелёный, как молодая трава */
            padding: 30px; /* Добавляем "воздух" по краям */
            font-family: 'Georgia', serif; /* Красивый классический шрифт */
        }

        /* 2. Раскрашиваем главный заголовок */

```

```
h1 {  
    color: #2e8b57; /* Цвет морской волны */  
    text-align: center; /* Размещаем по центру */  
    border-bottom: 3px solid #98fb98; /* Зелёная линия под заголовком */  
    padding-bottom: 15px; /* Отступ между текстом и линией */  
}  
  
/* 3. Раскрашиваем обычные абзацы */  
p {  
    color: #228b22; /* Лесной зелёный */  
    font-size: 18px; /* Чуть увеличиваем буквы */  
    line-height: 1.6; /* Делаем расстояние между строками побольше */  
}  
  
/* 4. Раскрашиваем подзаголовок */  
h2 {  
    color: #8b4513; /* Коричневый, как земля */  
    background-color: #ffffacd; /* Светло-жёлтый фон */  
    display: inline-block; /* Чтобы фон был только вокруг текста */  
    padding: 8px 15px; /* Отступы внутри жёлтого блока */  
    border-radius: 10px; /* Закругляем углы блока */  
}  
  
/* 5. Раскрашиваем список */  
ul {  
    color: #556b2f; /* Тёмный оливковый */  
    background-color: #fafad2; /* Светло-золотистый фон */  
    padding: 20px 20px 20px 40px; /* Отступы: сверху, справа, снизу, слева */  
    border-radius: 15px; /* Сильно закругляем углы */  
    border-left: 8px solid #daa520; /* Золотистая полоса слева */  
}  
  
li {  
    margin-bottom: 8px; /* Отступ между пунктами списка */  
    font-size: 17px;  
}  
  
/* 6. Особый стиль для последнего абзаца */  
p:last-child {  
    color: #b22222; /* Огненно-красный */  
    font-style: italic; /* Курсив */  
    text-align: right; /* Выравниваем по правому краю */  
    font-size: 20px; /* Делаем чуть крупнее */  
}
```

```
</style>
</head>
<body>
    <h1>Добро пожаловать в мой сад!</h1>

    <p>Сегодня я хочу рассказать вам о моих любимых цветах, которые растут на даче. Каждое утро, выходя в сад, я любуюсь этой красотой.</p>

    <p>Розы — мои самые любимые. У меня есть красные, розовые и белые сорта. Они пахнут так волшебно, что пчёлы прилетают даже с соседнего участка.</p>

    <h2>Что нужно цветам для счастья:</h2>

    <ul>
        <li>Солнечный свет</li>
        <li>Своевременный полив</li>
        <li>Лововь и забота</li>
    </ul>

    <p>Приезжайте в гости летом — покажу всё самое красивое!</p>

    <p>С теплом, Ваша соседка-садовод </p>
</body>
</html>
```

## 2.2. Сохраните и посмотрите результат

1. Нажмите **Ctrl+S** (Сохранить)
2. Перейдите в браузер с открытым файлом `сад.html`
3. Нажмите **кнопку F5** (или значок обновления 

**Волшебство!** Страница мгновенно преобразилась:

- Фон стал светло-зелёным
  - Заголовок — цвета морской волны с зелёной линией
  - Текст — лесного зелёного цвета
  - Подзаголовок на жёлтом фоне с закруглёнными углами
  - Список в золотистой рамке
  - Подпись красным курсивом справа
-

## ШАГ 3: Экспериментируем с мгновенными изменениями

Теперь начинается самое интересное! Мы будем менять значения и сразу видеть результат.

### Эксперимент 1: Меняем цвет фона

1. Вернитесь в редактор с файлом сад.html
2. Найдите строку: background-color: #f0ffff0;
3. Измените цвет на любой другой:

css

```
background-color: #f0f8ff; /* Голубой (AliceBlue) */
```

или

css

```
background-color: lavenderblush; /* Простое слово! */
```

4. Нажмите Ctrl+S
5. Перейдите в браузер и нажмите F5

**Фон страницы мгновенно изменился!**

### Эксперимент 2: Меняем цвет заголовка

1. Найдите строку: color: #2e8b57; (в разделе h1)
2. Измените на:

css

```
color: #8b0000; /* Тёмно-красный */
```

3. Сохраните (Ctrl+S) и обновите браузер (F5)

**Заголовок стал красным!**

### Эксперимент 3: Добавляем тень к заголовку

1. В разделе h1 добавьте новую строку:

css

`text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.3);`

2. Теперь раздел `h1` должен выглядеть так:

css

```
h1 {  
    color: #8b0000; /* Тёмно-красный */  
    text-align: center;  
    border-bottom: 3px solid #98fb98;  
    padding-bottom: 15px;  
    text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.3); /* НОВАЯ СТРОКА */  
}
```

3. Сохраните и обновите браузер

**Заголовок приобрел лёгкую тень, стал объёмным!**

## Эксперимент 4: Работаем с RGBA (прозрачность)

1. Найдите раздел `ul` (список)

2. Измените фон списка на полупрозрачный:

css

`background-color: rgba(250, 250, 210, 0.7); /* Последнее число 0.7 = 70% непрозрачности */`

3. Сохраните и обновите

**Фон списка стал полупрозрачным, сквозь него виден основной фон страницы!**

---

## ШАГ 4: Ваши собственные эксперименты

Попробуйте изменить следующие значения самостоятельно. После каждого изменения **не забывайте сохранять (Ctrl+S) и обновлять браузер (F5)**.

### Задание 1: Создайте «вечерний» вариант сада

- Фон страницы: #191970 (полуночно-синий)
- Цвет текста в абзацах: #ebebefa (лавандовый)

- Цвет заголовка: #ffd700 (золотой)
- Уберите границу под заголовком: border-bottom: none;

## Задание 2: Создайте «осенний» вариант

- Фон страницы: #ffffaf0 (цвет слоновой кости)
- Цвет заголовка: #d2691e (шоколадный)
- Фон подзаголовка (h2): #ffebcd (миндальный)
- Цвет последнего абзаца: #b8860b (тёмно-золотой)

## Задание 3: Поиграйте с простыми словами

Замените все HEX-коды на простые английские названия:

- #228b22 → forestgreen
  - #8b4513 → saddlebrown
  - #b22222 → firebrick
- 

## ШАГ 5: Создаём отдельный CSS-файл (профессиональный подход)

Давайте перенесём стили в отдельный файл, как это делают настоящие веб-дизайнеры.

### 5.1. Создайте новый файл style.css

1. В вашем редакторе создайте **новый файл** (Файл → Новый)
2. Скопируйте туда ВЕСЬ блок стилей из <style>...</style>, но без самих тегов <style>
3. Сохраните файл как style.css в ту же папку Практикум1\_Цвета

**Содержимое** style.css:

```
css
/*
СЕКРЕТНАЯ ЛАБОРАТОРИЯ ЦВЕТА */

/* 1. Меняем фон всей страницы */
```

```
body {  
    background-color: #f0ffff; /* Очень светлый зелёный, как молодая трава */  
    padding: 30px; /* Добавляем "воздух" по краям */  
    font-family: 'Georgia', serif; /* Красивый классический шрифт */  
}  
  
/* 2. Раскрашиваем главный заголовок */  
h1 {  
    color: #2e8b57; /* Цвет морской волны */  
    text-align: center; /* Размещаем по центру */  
    border-bottom: 3px solid #98fb98; /* Зелёная линия под заголовком */  
    padding-bottom: 15px; /* Отступ между текстом и линией */  
}  
  
/* 3. Раскрашиваем обычные абзацы */  
p {  
    color: #228b22; /* Лесной зелёный */  
    font-size: 18px; /* Чуть увеличиваем буквы */  
    line-height: 1.6; /* Делаем расстояние между строками побольше */  
}  
  
/* 4. Раскрашиваем подзаголовок */  
h2 {  
    color: #8b4513; /* Коричневый, как земля */  
    background-color: #ffffacd; /* Светло-жёлтый фон */  
    display: inline-block; /* Чтобы фон был только вокруг текста */  
    padding: 8px 15px; /* Отступы внутри жёлтого блока */  
    border-radius: 10px; /* Закругляем углы блока */  
}  
  
/* 5. Раскрашиваем список */  
ul {  
    color: #556b2f; /* Тёмный оливковый */  
    background-color: #fafad2; /* Светло-золотистый фон */  
    padding: 20px 20px 20px 40px; /* Отступы: сверху, справа, снизу, слева */  
    border-radius: 15px; /* Сильно закругляем углы */  
    border-left: 8px solid #daa520; /* Золотистая полоса слева */  
}  
  
li {  
    margin-bottom: 8px; /* Отступ между пунктами списка */  
    font-size: 17px;  
}
```

```
/* 6. Особый стиль для последнего абзаца */
p:last-child {
    color: #b22222; /* Огненно-красный */
    font-style: italic; /* Курсив */
    text-align: right; /* Выравниваем по правому краю */
    font-size: 20px; /* Делаем чуть крупнее */
}
```

## 5.2. Измените файл index.html

1. Откройте index.html
2. УДАЛИТЕ весь блок `<style>...</style>` (от `<style>` до `</style>` включительно)
3. Вместо него добавьте строку подключения CSS-файла:

html

```
<link rel="stylesheet" href="style.css">
```

4. Сохраните файл

## 5.3. Проверьте, что всё работает

Обновите страницу в браузере (F5). Вы должны увидеть ТО ЖЕ САМОЕ оформление, но теперь стили взяты из отдельного файла!

**Магия:** Теперь попробуйте изменить цвет фона в файле `style.css`, сохранить и обновить браузер. Всё работает точно так же!

---

## Что мы узнали и чему научились:

- ✓ **Создали HTML-страницу** с заголовками, абзацами и списком
- ✓ **Добавили стили** тремя разными способами (встроенные, в `<head>`, внешний файл)
- ✓ **Использовали разные способы задания цвета:** HEX-коды, простые слова, RGBA
- ✓ **Увидели мгновенный результат** после каждого изменения
- ✓ **Научились** менять: цвет текста, цвет фона, шрифт, отступы, границы, тени
- ✓ **Перенесли стили в отдельный файл** — профессиональный подход

**Ваша папка Практикум1\_Цвета теперь содержит:**

- index.html — структура страницы
- style.css — оформление страницы

## **Домашнее задание для закрепления:**

1. **Создайте новую страницу** recept.html **о вашем любимом блюде**
2. **Используйте отдельный CSS-файл** recept - style.css
3. **Раскрасьте:**
  - Заголовок рецепта — оранжевым (#ff8c00)
  - Ингредиенты — на светло-коричневом фоне (#f5deb3)
  - Способ приготовления — тёмно-зелёным текстом (#006400)
  - Совет от повара — розовым курсивом (#ff69b4)
4. **Поэкспериментируйте** с прозрачностью для фона ингредиентов

**Помните:** Самый лучший способ научиться — это экспериментировать. Не бойтесь пробовать странные сочетания цветов. Вы всегда можете нажать **Ctrl+Z** (отменить) или вернуть исходный код из этого урока.

**Вы только что совершили своё первое волшебство с CSS!** Вы из простого чёрно-белого текста создали красочную, живую страницу. И всё это — меняя всего несколько строк кода. В следующей главе мы научимся управлять текстом ещё тоньше — выбирать шрифты, размеры и начертания!

## ■ Цвет текста (color) vs цвет фона (background-color). Пробуем оба.

Дорогой друг, представьте, что вы пишете поздравительную открытку. Вы можете выбрать цвет чернил (это `color`) и выбрать цвет бумаги (это `background-color`). Оба важны, и вместе они создают настроение! Давайте разберёмся, как ими управлять.

### 1. Основное отличие — что есть что?

#### `color` — цвет текста (чернила)

css

```
p {  
    color: blue; /* Буквы будут синими */  
}
```

Это свойство меняет **цвет самого текста**, то есть чернил, которыми написаны буквы.

#### `background-color` — цвет фона (бумага)

css

```
p {  
    background-color: yellow; /* Под буквами будет жёлтый фон */  
}
```

Это свойство меняет **цвет области ПОД текстом**, то есть фон, на котором текст расположен.

---

### 2. Наглядный пример «Открытка»

Давайте создадим простой пример, чтобы увидеть разницу. Создайте файл `index.html`:

html

```
<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Открытка: color vs background-color</title>
    <style>
        /* Общие настройки для всех блоков */
        .блок {
            width: 300px;
            height: 150px;
            margin: 20px;
            padding: 20px;
            border: 2px solid #333;
            font-family: 'Georgia', serif;
            font-size: 20px;
            display: inline-block;
            vertical-align: top;
        }

        /* Просто текст по умолчанию */
        .пример1 {
            color: black;
            background-color: white;
        }

        /* Меняем ТОЛЬКО цвет текста */
        .пример2 {
            color: darkred;      /* Текст бордовый */
            background-color: white; /* Фон остаётся белым */
        }

        /* Меняем ТОЛЬКО цвет фона */
        .пример3 {
            color: black;      /* Текст чёрный */
            background-color: lightblue; /* Фон голубой */
        }

        /* Меняем ОБА цвета */
        .пример4 {
            color: white;      /* Текст белый */
            background-color: darkgreen; /* Фон тёмно-зелёный */
        }
    </style>

```

```
}

/* Плохое сочетание - текст плохо читается */
.пример5 {
    color: lightgray; /* Светло-серый текст */
    background-color: white; /* На белом фоне */
}

/* Хорошее сочетание с прозрачностью */
.пример6 {
    color: navy; /* Тёмно-синий текст */
    background-color: rgba(255, 228, 196, 0.7); /* Полупрозрачный бисквитный фон */
}

</style>
</head>
<body>
    <h1>Разница между color и background-color</h1>

    <div class="блок пример1">
        <strong>1. По умолчанию:</strong><br>
        color: black<br>
        background-color: white<br>
        Как обычная бумага
    </div>

    <div class="блок пример2">
        <strong>2. Только color:</strong><br>
        color: darkred<br>
        background-color: white<br>
        Красные чернила на белой бумаге
    </div>

    <div class="блок пример3">
        <strong>3. Только background-color:</strong><br>
        color: black<br>
        background-color: lightblue<br>
        Чёрные чернила на голубой бумаге
    </div>

    <div class="блок пример4">
        <strong>4. Оба цвета:</strong><br>
        color: white<br>
        background-color: darkgreen<br>
        Белые чернила на зелёной бумаге
    </div>
```

```

</div>

<div class="блок пример5">
    <strong>5. Плохая читаемость:</strong><br>
    color: lightgray<br>
    background-color: white<br>
    Текст почти не видно!
</div>

<div class="блок пример6">
    <strong>6. С прозрачностью:</strong><br>
    color: navy<br>
    background-color: rgba(255,228,196,0.7)<br>
    Видно фон страницы сквозь блок
</div>

<p style="margin-top: 30px; color: #666;">
    <strong>Вывод:</strong> color = цвет текста, background-color = цвет фона под
текстом.<br>
    Важно подбирать их так, чтобы текст хорошо читался!
</p>
</body>
</html>

```

**Сохраните файл и откройте в браузере.** Вы увидите 6 разных комбинаций цветов текста и фона.

## Разница между **color** и **background-color**

**1. По умолчанию:**  
**color: black**  
**background-color: white**  
 Как обычная бумага

**2. Только **color**:**  
**color: darkred**  
**background-color: white**  
 Красные чернила на белой бумаге

**3. Только **background-color**:**  
**color: black**  
**background-color: lightblue**  
 Чёрные чернила на голубой бумаге

**4. Оба цвета:**  
**color: white**  
**background-color: darkgreen**  
 Белые чернила на зелёной бумаге

**5. Плохая читаемость:**  
**color: lightgray**  
**background-color: white**  
 Текст почти не видно!

**6. С прозрачностью:**  
**color: navy**  
**background-color:**  
**rgba(255,228,196,0.7)**  
 Видно фон страницы сквозь блок

Вывод: color = цвет текста, background-color = цвет фона под текстом.  
 Важно подбирать их так, чтобы текст хорошо читался!

---

### 3. Практикум: Создаём «Набор открыток»

Давайте создадим более практичный пример. Создайте файл index.html:

html

```
<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Мои поздравительные открытки</title>
    <style>
        /* Общие настройки страницы */
        body {
            background-color: #f8f8f8;
            font-family: 'Arial', sans-serif;
            padding: 30px;
        }

        h1 {
            text-align: center;
            color: #333;
            margin-bottom: 40px;
        }

        /* Контейнер для открыток */
        .открытки {
            display: flex;
            flex-wrap: wrap;
            gap: 25px;
            justify-content: center;
        }

        /* Общий стиль для каждой открытки */
        .открытка {
            width: 280px;
            height: 380px;
            padding: 25px;
            border-radius: 15px;
            box-shadow: 0 8px 20px rgba(0,0,0,0.1);
        }
    </style>
</head>
<body>
    <h1>Мои поздравительные открытки</h1>
    <div class="открытки">
        <div class="открытка"></div>
        <div class="открытка"></div>
        <div class="открытка"></div>
        <div class="открытка"></div>
        <div class="открытка"></div>
        <div class="открытка"></div>
    </div>
</body>
</html>
```

```
display: flex;
flex-direction: column;
justify-content: space-between;
}

/* ОТКРЫТКА 1: День рождения */
.день-рождения {
    color: #8b0000; /* ТЁМНО-КРАСНЫЙ текст */
    background-color: #fffacd; /* СВЕТЛО-ЖЁЛТЫЙ фон */
    border: 5px dashed #ffd700; /* ЗОЛОТАЯ пунктирная рамка */
}

/* ОТКРЫТКА 2: 8 Марта */
.весна {
    color: #ff1493; /* ЯРКО-РОЗОВЫЙ текст */
    background-color: #f0fff0; /* СВЕТЛО-ЗЕЛЁНЫЙ фон */
    border: 3px dotted #32cd32; /* ЗЕЛЁНЫЙ точечный бордюр */
}

/* ОТКРЫТКА 3: Новый год */
.новый-год {
    color: #ffffff; /* БЕЛЫЙ текст */
    background-color: #006400; /* ТЁМНО-ЗЕЛЁНЫЙ фон */
    border: 8px double #ffd700; /* ЗОЛОТАЯ двойная рамка */
}

/* ОТКРЫТКА 4: Соболезнование */
.соболезнование {
    color: #2f4f4f; /* ТЁМНО-СЛАТОВЫЙ текст */
    background-color: #f5f5f5; /* ОЧЕНЬ СВЕТЛО-СЕРЫЙ фон */
    border: 1px solid #808080; /* СЕРАЯ тонкая рамка */
}

/* ОТКРЫТКА 5: Любовная записка */
.любовь {
    color: #800080; /* ФИОЛЕТОВЫЙ текст */
    background: linear-gradient(to bottom, #ffb6c1, #ffe4e1); /* РОЗОВЫЙ градиентный фон */
    border: 2px solid #db7093; /* ТЁМНО-РОЗОВАЯ рамка */
}

/* Стили для текста внутри открыток */
.заголовок {
    font-size: 24px;
```

```
        font-weight: bold;
        text-align: center;
        margin-bottom: 15px;
    }

    .текст {
        font-size: 18px;
        line-height: 1.5;
        flex-grow: 1;
    }

    .подпись {
        font-style: italic;
        text-align: right;
        margin-top: 20px;
        font-size: 16px;
    }

/* Панель управления */
.управление {
    background-color: white;
    padding: 20px;
    margin-top: 40px;
    border-radius: 10px;
    box-shadow: 0 4px 10px rgba(0,0,0,0.1);
}

.кнопка {
    padding: 10px 20px;
    margin: 5px;
    border: none;
    border-radius: 5px;
    cursor: pointer;
    font-size: 16px;
    transition: all 0.3s;
}

.кнопка:hover {
    transform: translateY(-2px);
    box-shadow: 0 4px 8px rgba(0,0,0,0.2);
}

</style>
</head>
<body>
```

```
<h1> Коллекция поздравительных открыток</h1>
```

```
<div class="открытки">
```

```
    <!-- Открытка 1 -->
```

```
    <div class="открытка день-рождения" id="открытка1">
```

```
        <div class="заголовок">С Днём Рождения!</div>
```

```
        <div class="текст">
```

Дорогой друг! Желаю тебе здоровья, счастья, радости и множества прекрасных моментов в жизни. Пусть каждый день приносит улыбки!

```
    </div>
```

```
    <div class="подпись">С любовью, Мария</div>
```

```
</div>
```

```
    <!-- Открытка 2 -->
```

```
    <div class="открытка весна" id="открытка2">
```

```
        <div class="заголовок">С 8 Марта!</div>
```

```
        <div class="текст">
```

Милая женщина, поздравляю с праздником весны! Желаю всегда оставаться такой же прекрасной, нежной и удивительной.

```
    </div>
```

```
    <div class="подпись">Твой друг, Сергей</div>
```

```
</div>
```

```
    <!-- Открытка 3 -->
```

```
    <div class="открытка новый-год" id="открытка3">
```

```
        <div class="заголовок">С Новым Годом!</div>
```

```
        <div class="текст">
```

Пусть новый год принесёт в ваш дом счастье, уют и благополучие. Пусть сбудутся все самые заветные мечты!

```
    </div>
```

```
    <div class="подпись">Ваша семья Ивановых</div>
```

```
</div>
```

```
    <!-- Открытка 4 -->
```

```
    <div class="открытка соболезнование" id="открытка4">
```

```
        <div class="заголовок">Соболезну...</div>
```

```
        <div class="текст">
```

Примите мои искренние соболезнования. Скорблю вместе с вами. Крепитесь, дорогие.

```
    </div>
```

```
    <div class="подпись">С уважением, Анна Петровна</div>
```

```
</div>
```

```
    <!-- Открытка 5 -->
```

```

<div class="открытка любовь" id="открытка5">
    <div class="заголовок">Любимому</div>
    <div class="текст">
        Каждый день с тобой — это счастье. Спасибо, что ты есть в моей жизни. Люблю
        тебя больше всего на свете.
    </div>
    <div class="подпись">Навсегда твоя</div>
</div>
</div>

<!-- Панель управления для экспериментов -->
<div class="управление">
    <h2>Лаборатория цвета: Поэкспериментируйте!</h2>
    <p>Выберите открытку и поменяйте её цвета:</p>

    <div>
        <strong>Открытка:</strong>
        <button class="кнопка" onclick="выбратьОткрытку(1)">🎂 День рождения</button>
        <button class="кнопка" onclick="выбратьОткрытку(2)">🌸 8 Марта</button>
        <button class="кнопка" onclick="выбратьОткрытку(3)">🎄 Новый год</button>
        <button class="кнопка" onclick="выбратьОткрытку(4)">● Соболезнование</button>
        <button class="кнопка" onclick="выбратьОткрытку(5)">❤️ Любовь</button>
    </div>

    <div style="margin-top: 20px;">
        <strong>Цвет текста (color):</strong>
        <button class="кнопка" style="background-color: #000000; color: white;">
            onclick="изменитьЦветТекста('#000000')">Чёрный</button>
        <button class="кнопка" style="background-color: #8b0000; color: white;">
            onclick="изменитьЦветТекста('#8b0000')">Бордовый</button>
        <button class="кнопка" style="background-color: #00008b; color: white;">
            onclick="изменитьЦветТекста('#00008b')">Тёмно-синий</button>
        <button class="кнопка" style="background-color: #006400; color: white;">
            onclick="изменитьЦветТекста('#006400')">Тёмно-зелёный</button>
        <button class="кнопка" style="background-color: #4b0082; color: white;">
            onclick="изменитьЦветТекста('#4b0082')">Индиго</button>
    </div>

    <div style="margin-top: 20px;">
        <strong>Цвет фона (background-color):</strong>
        <button class="кнопка" style="background-color: #ffffff;">
            onclick="изменитьЦветФона('#ffffff')">Белый</button>
        <button class="кнопка" style="background-color: #fffacd;">
            onclick="изменитьЦветФона('#fffacd')">Светло-жёлтый</button>
    </div>

```

```

<button class="кнопка" style="background-color: #f0fff0;"  

    onclick="изменитьЦветФона('#f0fff0')>Светло-зелёный</button>  

    <button class="кнопка" style="background-color: #f0f8ff;"  

    onclick="изменитьЦветФона('#f0f8ff')>Голубой</button>  

    <button class="кнопка" style="background-color: #fff0f5;"  

    onclick="изменитьЦветФона('#fff0f5')>Лавандовый</button>  

</div>  
  

<div style="margin-top: 20px;">  

    <strong>Быстрые пресеты:</strong>  

    <button class="кнопка" style="background-color: #800000; color: white;"  

        onclick="применитьПресет('#800000', '#ffdab9')>🍓 Клубника</button>  

    <button class="кнопка" style="background-color: #2e8b57; color: white;"  

        onclick="применитьПресет('#2e8b57', '#f5ffa')>🌿 Мята</button>  

    <button class="кнопка" style="background-color: #191970; color: white;"  

        onclick="применитьПресет('#191970', '#e6e6fa')>🌙 Ночь</button>  

    <button class="кнопка" style="background-color: #8b4513; color: white;"  

        onclick="применитьПресет('#8b4513', '#faebd7')>☕ Корица</button>  

</div>  
  

<div id="текущаяОткрытка" style="margin-top: 20px; padding: 15px; background-color: #eef;  

border-radius: 5px;">  

    <strong>Выбрана открытка:</strong> <span id="номерОткрытки">1 (День  

рождения)</span><br>  

    <strong>Текущий color:</strong> <span id="текущийColor">#8b0000</span><br>  

    <strong>Текущий background-color:</strong> <span id="текущийФон">#ffffacd</span>  

</div>  

</div>  
  

<script>  

    let выбраннаяОткрытка = 1;  
  

    function выбратьОткрытку(номер) {  

        выбраннаяОткрытка = номер;  

        const открытка = document.getElementById('открытка${номер}');  

        const стили = window.getComputedStyle(открытка);  
  

        // Обновляем информацию  

        document.getElementById('номерОткрытки').textContent = `${номер} ($  

{открытка.querySelector('.заголовок').textContent});  

        document.getElementById('текущийColor').textContent = стили.color;  

        document.getElementById('текущийФон').textContent = стили.backgroundColor;  
  

        // Подсвечиваем выбранную открытку

```

```
document.querySelectorAll('.открытка').forEach(откр => {
    откр.style.boxShadow = '0 8px 20px rgba(0,0,0,0.1)';
});
открытка.style.boxShadow = '0 0 0 3px #4a90e2, 0 8px 20px rgba(0,0,0,0.2)';
}

function изменитьЦветТекста(цвет) {
    const открытка = document.getElementById(`открытка${выбраннаяОткрытка}`);
    открытка.style.color = цвет;
    document.getElementById('текущийColor').textContent = цвет;
}

function изменитьЦветФона(цвет) {
    const открытка = document.getElementById(`открытка${выбраннаяОткрытка}`);
    открытка.style.backgroundColor = цвет;
    document.getElementById('текущийФон').textContent = цвет;
}

function применитьПресет(цветТекста, цветФона) {
    изменитьЦветТекста(цветТекста);
    изменитьЦветФона(цветФона);
}

// Инициализация
window.onload = function() {
    выбратьОткрытку(1);
};

</script>
</body>
</html>
```

 Коллекция поздравительных открыток

**С Днём Рождения!**

Дорогой друг! Желаю тебе здоровья, счастья, радости и множества прекрасных моментов в жизни. Пусть каждый день приносит улыбки!

С любовью, Мария

**С 8 Марта!**

Милая женщина, поздравляю с праздником весны! Желаю всегда оставаться такой же прекрасной, нежной и удивительной.

Твой друг, Сергей

**С Новым Годом!**

Пусть новый год принесёт в ваш дом счастье, уют и благополучие. Пусть сбудутся все самые заветные мечты!

Ваша семья Ивановых

**Соболезнование...**

Примите мои искренние соболезнования. Скорблю вместе с вами. Крепитесь, дорогие.

С уважением, Анна Петровна

**Любимому**

Каждый день с тобой — это счастье. Спасибо, что ты есть в моей жизни. Люблю тебя больше всего на свете.

Навсегда твоя

**Лаборатория цвета: Позэкспериментируйте!**

Выберите открытку и поменяйте её цвета:

Открытка:  День рождения  8 Марта  Новый год  Соболезнование  Любовь

Цвет текста (color): Чёрный Бордовый Тёмно-синий Тёмно-зелёный Индigo

Цвет фона (background-color): Белый Светло-жёлтый Светло-зелёный Голубой Лавандовый

Быстрые пресеты: Клубника Мата Ночь Корица

Выбрана открытка: 5 (Любимому)  
Текущий color: #800000  
Текущий background-color: #fdab9

## 4. Что мы можем сделать с этим файлом?

- Сохраните** как index.html и откройте в браузере
- Нажмите на кнопки** в строке «Открытка:» открытками, чтобы выбрать открытку
- Экспериментируйте** с цветами текста и фона
- Нажимайте** «Быстрые пресеты» для красивых сочетаний

### Особенности этого примера:

- Все открытки используют разные комбинации color и background-color
- Есть интерактивная панель для экспериментов
- Можно видеть, как меняются цвета в реальном времени
- Показаны как хорошие, так и плохие сочетания

## 5. Важные правила сочетания цветов

### ХОРОШИЕ сочетания (текст хорошо читается):

css

```
/* Тёмный текст на светлом фоне */
color: black;
background-color: white;
```

```
/* Белый текст на тёмном фоне */
```

```
color: white;
```

```
background-color: darkblue;
```

```
/* Контрастные сочетания */
```

```
color: darkred;
```

```
background-color: lightyellow;
```

```
color: darkgreen;
```

```
background-color: #f0ffff;
```

## ПЛОХИЕ сочетания (текст плохо читается):

css

```
/* Слишком близкие цвета */
```

```
color: lightgray;
```

```
background-color: white;
```

```
color: yellow;
```

```
background-color: lightyellow;
```

```
/* Очень яркие сочетания */
```

```
color: brightpink;
```

```
background-color: brightgreen; /* Режет глаза! */
```

## Правило контраста:

Для основного текста старайтесь, чтобы разница между цветом текста и фона была **максимальной**. Тёмное на светлом или светлое на тёмном.

---

## 6. Эксперимент: Создайте свою открытку

Создайте новый файл `index.html`:

html

```
<!DOCTYPE html>
```

```
<html lang="ru">
```

```
<head>
    <meta charset="UTF-8">
    <title>Моя авторская открытка</title>
    <style>

        body {
            background-color: #f5f5f5;
            padding: 40px;
            font-family: 'Georgia', serif;
        }

        .открытка {
            width: 400px;
            height: 500px;
            margin: 0 auto;
            padding: 40px;
            border-radius: 20px;
            box-shadow: 0 15px 35px rgba(0,0,0,0.2);

            /* ИЗМЕНЯЙТЕ ЭТИ ДВЕ СТРОЧКИ! */
            color: #2c3e50; /* Цвет текста (попробуйте разные) */
            background-color: #ecf0f1; /* Цвет фона (попробуйте разные) */
        }

        h1 {
            text-align: center;
            margin-bottom: 30px;
            font-size: 32px;
        }

        .текст {
            font-size: 20px;
            line-height: 1.7;
            margin-bottom: 30px;
        }

        .подпись {
            text-align: right;
            font-style: italic;
            font-size: 22px;
        }

        .управление {
            background: white;
            padding: 20px;
        }
    </style>

```

```
margin-top: 30px;  
border-radius: 10px;  
}  
  
input {  
    margin: 10px;  
    padding: 8px;  
    width: 120px;  
}  
</style>  
</head>  
<body>  
    <h1>🎨 Создай свою открытку</h1>  
  
    <div class="открытка" id="мояОткрытка">  
        <h1 id="заголовок">Дорогая мама!</h1>  
        <div class="текст" id="основнойТекст">  
            Хочу сказать тебе, как я тебя люблю и ценю. Ты — самый важный человек в моей  
жизни. Спасибо за всё, что ты для меня делаешь каждый день.  
        </div>  
        <div class="подпись" id="подпись">Твоя дочь, Катя</div>  
    </div>  
  
    <div class="управление">  
        <h3>Редактор цветов:</h3>  
  
        <div>  
            <label>Цвет текста (color): </label>  
            <input type="color" id="цветТекста" value="#2c3e50" onchange="обновитьЦвета()">  
            <span id="кодТекста">#2c3e50</span>  
        </div>  
  
        <div>  
            <label>Цвет фона (background-color): </label>  
            <input type="color" id="цветФона" value="#ecf0f1" onchange="обновитьЦвета()">  
            <span id="кодФона">#ecf0f1</span>  
        </div>  
  
        <div style="margin-top: 20px;">  
            <button onclick="сохранитьОткрытку()">💾 Сохранить открытку</button>  
            <button onclick="случайныеЦвета()">🎲 Случайные цвета</button>  
        </div>  
    </div>
```

```
<script>

function обновитьЦвета() {
    const цветТекста = document.getElementById('цветТекста').value;
    const цветФона = document.getElementById('цветФона').value;
    const открытка = document.getElementById('мояОткрытка');

    открытка.style.color = цветТекста;
    открытка.style.backgroundColor = цветФона;

    document.getElementById('кодТекста').textContent = цветТекста;
    document.getElementById('кодФона').textContent = цветФона;
}

function случайныеЦвета() {
    // Генерируем случайные цвета
    const случайныйТекст = '#' + Math.floor(Math.random()*16777215).toString(16);
    const случайныйФон = '#' + Math.floor(Math.random()*16777215).toString(16);

    document.getElementById('цветТекста').value = случайныйТекст;
    document.getElementById('цветФона').value = случайныйФон;

    обновитьЦвета();
}

function сохранитьОткрытку() {
    alert('Открытка сохранена! Можно сделать скриншот (Print Screen)');
}

// Инициализация
обновитьЦвета();

</script>
</body>
</html>
```

 Создай свою открытку

Дорогая мама!

Хочу сказать тебе, как я тебя люблю и ценю.  
Ты — самый важный человек в моей жизни.  
Спасибо за всё, что ты для меня делаешь  
каждый день.

Твоя дочь, Катя

**Редактор цветов:**

Цвет текста (color):  #ca1664

Цвет фона (background-color):  #ffcc2d

### Что можно делать с этой открыткой:

1. Менять цвет текста и фона с помощью цветовых полей
2. Нажимать «Случайные цвета» для неожиданных сочетаний
3. Делать скриншот (Print Screen), чтобы сохранить результат
4. Изменять текст прямо в редакторе кода

## 7. Краткие итоги и домашнее задание

### Что мы узнали:

- ✓ `color` — меняет цвет самого текста (чернила)
- ✓ `background-color` — меняет цвет фона под текстом (бумага)
- ✓ Их можно использовать **отдельно или вместе**
- ✓ Важно подбирать **контрастные сочетания** для читаемости
- ✓ Можно использовать **простые слова, HEX-коды, RGB/RGBA**
- ✓ **Интерактивные элементы** позволяют экспериментировать вживую

## **Домашнее задание:**

### **Задание 1: «Шахматная доска»**

Создайте страницу с 8×8 клетками, используя чередующиеся background-color:

- Чёрные клетки: #769656
- Белые клетки: #eeeeed2

### **Задание 2: «Радужный текст»**

Создайте 7 абзацев с названиями цветов радуги, где каждый абзац имеет:

- Соответствующий color (красный, оранжевый, жёлтый...)
- Светлый background-color того же оттенка

### **Задание 3: «Проверка читаемости»**

Создайте 3 варианта оформления одного текста:

1. Отлично читаемый (сильный контраст)
2. Плохо читаемый (малый контраст)
3. Нечитаемый (почти одинаковые цвета)

### **Задание 4: «Создай открытку для друга»**

Используя файл index.html:

1. Измените текст на поздравительный для друга
2. Подберите цвета, которые нравятся вашему другу
3. Сделайте скриншот и отправьте другу

## **Золотое правило:**

Всегда проверяйте, хорошо ли читается текст. Можно использовать простое правило: **если прищурившись текст сливаётся с фоном — нужно усилить контраст.**

**Поздравляю!** Теперь вы мастер цвета в CSS! Вы умеете управлять и текстом, и фоном, создавая красивые и удобочитаемые композиции. В следующей главе мы научимся выбирать шрифты — это будет не менее увлекательно! 

## ● Глава 3: Учимся управлять ТЕКСТОМ.

### ■ Выбираем шрифт (font-family): Как найти красивые и понятные шрифты?

Дорогой друг, представьте, что вы выбираете наряд для важного мероприятия. Шрифт — это «одежда» для вашего текста! Какой шрифт надеть? Строгий костюм (Times New Roman) или удобную домашнюю одежду (Comic Sans)? Давайте научимся выбирать шрифты с умом и вкусом.

## 1. Что такое font-family и как она работает?

font-family — это свойство CSS, которое говорит браузеру: «Покажи этот текст таким-то шрифтом».

css

```
p {  
    font-family: "Times New Roman", Times, serif;  
}
```

### Важная особенность: цепочка шрифтов

Браузер читает список слева направо:

1. Пробует использовать "**Times New Roman**"
2. Если его нет у пользователя → пробует **Times**
3. Если и его нет → использует любой **шрифт с засечками (serif)**

**Почему так?** Не все компьютеры имеют одинаковые шрифты! Поэтому мы даём браузеру варианты на выбор.

---

## 2. Основные категории шрифтов (как стили одежды)

### A. Шрифты С ЗАСЕЧКАМИ (Serif) — «Классические костюмы»

У этих шрифтов есть маленькие чёрточки на концах букв. Выглядят традиционно, официально, серьёзно.

css

```
font-family: "Times New Roman", Times, serif;  
font-family: Georgia, serif;  
font-family: "Palatino Linotype", "Book Antiqua", Palatino, serif;
```

**Где использовать:** Книги, газеты, официальные документы, блоги о классике.

**Примеры:**

text

Times New Roman — Строгий, официальный  
Georgia — Элегантный, удобный для чтения с экрана  
Garamond — Изящный, старинный

### B. Шрифты БЕЗ ЗАСЕЧЕК (Sans-serif) — «Современная одежда»

Чистые, без лишних деталей. Выглядят современно, минималистично, дружелюбно.

css

```
font-family: Arial, Helvetica, sans-serif;  
font-family: "Trebuchet MS", sans-serif;  
font-family: Verdana, Geneva, sans-serif;
```

**Где использовать:** Веб-сайты, интерфейсы, презентации, современный дизайн.

**Примеры:**

text

Arial — Универсальный, нейтральный  
Verdana — Широкий, очень читаемый на экране  
Helvetica — Дизайнерский, стильный

## **С. МОНОШИРИННЫЕ (Monospace) — «Пишущая машинка»**

Все буквы одинаковой ширины. Как текст на старой пишущей машинке или в командной строке.

css

```
font-family: "Courier New", Courier, monospace;  
font-family: "Lucida Console", Monaco, monospace;
```

**Где использовать:** Код программирования, техническая документация, имитация печатной машинки.

**Примеры:**

text

Courier New — Классическая пишущая машинка  
Consolas — Современный, удобный для чтения кода

## **D. РУКОПИСНЫЕ (Cursive) — «Почерк от руки»**

Имитируют рукописный текст. Красивые, но могут быть плохо читаемы.

css

```
font-family: "Comic Sans MS", cursive;  
font-family: "Brush Script MT", cursive;
```

**Где использовать:** Приглашения, открытки, детские сайты (осторожно!).

**Примеры:**

text

Comic Sans MS — Детский, неформальный (очень спорный!)  
Brush Script — Элегантный курсив

## **E. ДЕКОРАТИВНЫЕ (Fantasy) — «Карнавальные костюмы»**

Очень стилизованные, художественные. Часто тематические.

css

```
font-family: Impact, Charcoal, fantasy;  
font-family: Papyrus, fantasy;
```

**Где использовать:** Заголовки, логотипы, специальные проекты (очень редко!).

**Примеры:**

text

Impact — Жирный, громкий (для заголовков)

Papyrus — «Древний» стиль

---

### 3. Безопасные веб-шрифты (есть на всех компьютерах)

Вот список шрифтов, которые точно есть почти на каждом компьютере:

css

*/\* БЕЗ ЗАСЕЧЕК (самые безопасные) \*/*

**font-family:** Arial, sans-serif;  
**font-family:** Helvetica, sans-serif;  
**font-family:** Verdana, sans-serif;  
**font-family:** Tahoma, sans-serif;  
**font-family:** Trebuchet MS, sans-serif;

*/\* С ЗАСЕЧКАМИ \*/*

**font-family:** Times New Roman, serif;  
**font-family:** Georgia, serif;  
**font-family:** Palatino, serif;

*/\* МОНОШИРИННЫЕ \*/*

**font-family:** Courier New, monospace;  
**font-family:** Lucida Console, monospace;

---

### 4. Как подключить красивые шрифты из интернета?

Вот где начинается настоящее волшебство! Вы можете использовать тысячи красивых шрифтов бесплатно.

# Способ 1: Google Fonts (самый простой и популярный)

## Шаг 1: Идём на сайт

Откройте в браузере: [fonts.google.com](https://fonts.google.com)

## Шаг 2: Выбираем шрифт

- В поиске можно ввести русские слова: «красивый», «элегантный», «рукописный»
- Или смотреть категории: Serif, Sans-serif, Display, Handwriting
- Нажмите на понравившийся шрифт

## Шаг 3: Выбираем начертания

Нажмите «Select this style» для нужных вариантов:

- **Regular** — обычный
- **Italic** — курсив
- **Bold** — жирный
- **Light** — тонкий

## Шаг 4: Копируем код

В правой панели появятся два блока кода:

### 1. Для <head> (подключение шрифта):

```
html
<link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link href="https://fonts.googleapis.com/css2?family=Roboto:wght@400;700&display=swap"
rel="stylesheet">
```

### 2. Для CSS (использование шрифта):

```
css
font-family: 'Roboto', sans-serif;
```

## Шаг 5: Используем в своём проекте

Создайте файл `шрифты.html`:

```
html
<!DOCTYPE html>
```

```
<html lang="ru">
<head>
    <meta charset="UTF-8">
    <title>Эксперименты с Google Fonts</title>

    <!-- Подключаем шрифты из Google Fonts -->
    <link rel="preconnect" href="https://fonts.googleapis.com">
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
    <link href="https://fonts.googleapis.com/css2?
family=Cormorant+Garamond:wght@400;700&family=Dancing+Script:wght@400;700&family=Roboto:wg
ht@300;400;700&display=swap" rel="stylesheet">

<style>
    body {
        font-family: 'Roboto', sans-serif;
        padding: 40px;
        background-color: #f8f8f8;
        max-width: 800px;
        margin: 0 auto;
    }

    h1 {
        font-family: 'Cormorant Garamond', serif;
        font-size: 48px;
        color: #2c3e50;
        text-align: center;
        margin-bottom: 40px;
    }

    .рукописный {
        font-family: 'Dancing Script', cursive;
        font-size: 28px;
        color: #8b0000;
        text-align: center;
        margin: 30px 0;
    }

    .разные-стили {
        background: white;
        padding: 25px;
        border-radius: 15px;
        margin: 20px 0;
        box-shadow: 0 5px 15px rgba(0,0,0,0.1);
    }
}
```

```
.обычный { font-weight: 300; }
.средний { font-weight: 400; }
.жирный { font-weight: 700; }
.курсив { font-style: italic; }

.пример-шрифта {
    margin: 25px 0;
    padding: 20px;
    border-left: 5px solid;
}

.георгия {
    font-family: Georgia, serif;
    border-left-color: #3498db;
}

.ариал {
    font-family: Arial, sans-serif;
    border-left-color: #2ecc71;
}

.курьер {
    font-family: 'Courier New', monospace;
    border-left-color: #e74c3c;
}

.КОМИК {
    font-family: 'Comic Sans MS', cursive;
    border-left-color: #f39c12;
}

.папирус {
    font-family: Papyrus, fantasy;
    border-left-color: #9b59b6;
}

</style>
</head>
<body>
    <h1>Галерея шрифтов</h1>

    <div class="рукописный">
        «Типографика — это искусство, доступное каждому»
    </div>
```

```
<div class="разные-стили">
  <h2>Один шрифт — разные начертания (Roboto):</h2>
  <p class="обычный">Это лёгкое начертание (300) — нежное и воздушное</p>
  <p class="средний">Это обычное начертание (400) — стандартный вес</p>
  <p class="жирный">Это жирное начертание (700) — для акцентов</p>
  <p class="курсив">А это курсив — для выделения мыслей</p>
</div>
```

<h2>Сравнение основных категорий:</h2>

```
<div class="пример-шрифта георгия">
  <strong>Serif (с засечками) — Georgia:</strong><br>
  Засечки помогают глазу скользить по строке. Идеально для длинных текстов, книг, статей. Выглядит традиционно и надёжно.
</div>
```

```
<div class="пример-шрифта ариал">
  <strong>Sans-serif (без засечек) — Arial:</strong><br>
  Чистый, современный, нейтральный. Отлично подходит для интерфейсов, веб-сайтов, экранного чтения. Не отвлекает от содержания.
</div>
```

```
<div class="пример-шрифта курьер">
  <strong>Monospace (моноширинный) — Courier New:</strong><br>
  Все буквы одинаковой ширины. Используется для программирования, командной строки, имитации печатной машинки. Создаёт техническое ощущение.
</div>
```

```
<div class="пример-шрифта комик">
  <strong>Cursive (рукописный) — Comic Sans MS:</strong><br>
  Неформальный, дружелюбный, «детский». Хорош для открыток, приглашений, но плох для серьёзных текстов. (Дизайнеры его не любят!)
</div>
```

```
<div class="пример-шрифта папирус">
  <strong>Fantasy (декоративный) — Papyrus:</strong><br>
  Тематический, художественный. Используйте крайне осторожно и только для специальных целей! Часто выглядит безвкусно.
</div>
```

```
<div style="margin-top: 40px; padding: 20px; background: #2c3e50; color: white; border-radius: 10px;">
  <h3>Правила хорошего тона:</h3>
```

```
<ul>
    <li>Используйте не более 2-3 шрифтов на одной странице</li>
    <li>Для основного текста — простые, читаемые шрифты</li>
    <li>Для заголовков — можно более выразительные</li>
    <li>Проверяйте читаемость на разных устройствах</li>
    <li>Учитывайте настроение вашего проекта</li>
</ul>
</div>
</body>
</html>
```

## Способ 2: Шрифты Windows vs Mac vs Linux

Проблема: некоторые шрифты есть только на Windows, другие — только на Mac.

Решение — указывать альтернативы:

css

```
/* Хорошая практика для веба */
body {
    font-family: -apple-system, BlinkMacSystemFont, "Segoe UI", Roboto,
        "Helvetica Neue", Arial, sans-serif, "Apple Color Emoji",
        "Segoe UI Emoji", "Segoe UI Symbol";
}

/* Для русского текста */
.русский-текст {
    font-family: 'Roboto', 'Helvetica Neue', Arial, sans-serif;
}
```

---

## 5. Практикум: Создаём «Шрифтовую книгу»

Создайте файл `шрифтовая-книга.html`:

html

```
<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="UTF-8">
    <title>Моя шрифтовая книга</title>
```

```
<!-- Подключаем 5 разных шрифтов из Google Fonts -->
<link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link href="https://fonts.googleapis.com/css2?
family=Caveat:wght@400;700&family=Lora:wght@400;700&family=Montserrat:wght@300;400;700&famil
y=Open+Sans:wght@300;400;700&family=Playfair+Display:wght@400;700&display=swap"
rel="stylesheet">

<style>
  * {
    box-sizing: border-box;
  }

  body {
    background-color: #f5f7fa;
    padding: 30px;
    font-family: 'Open Sans', sans-serif;
    color: #333;
    line-height: 1.6;
  }

  .контейнер {
    max-width: 1000px;
    margin: 0 auto;
    background: white;
    padding: 40px;
    border-radius: 20px;
    box-shadow: 0 10px 30px rgba(0,0,0,0.1);
  }

  h1 {
    text-align: center;
    color: #2c3e50;
    margin-bottom: 40px;
    font-size: 42px;
  }

  .шрифт-карточка {
    margin: 35px 0;
    padding: 25px;
    border-radius: 15px;
    transition: all 0.3s ease;
    border-left: 8px solid;
  }
}
```

```
.шрифт-карточка:hover {
    transform: translateY(-5px);
    box-shadow: 0 10px 25px rgba(0,0,0,0.15);
}

.название-шрифта {
    font-size: 32px;
    margin-bottom: 15px;
    padding-bottom: 10px;
    border-bottom: 2px solid;
}

.описание {
    font-size: 16px;
    color: #666;
    margin-bottom: 20px;
    font-style: italic;
}

.примеры {
    display: flex;
    flex-wrap: wrap;
    gap: 20px;
    margin-top: 20px;
}

.пример {
    flex: 1;
    min-width: 200px;
    padding: 15px;
    border-radius: 8px;
    background: #f8f9fa;
}

.вес-300 { font-weight: 300; }
.вес-400 { font-weight: 400; }
.вес-700 { font-weight: 700; }
.курсив { font-style: italic; }

/* ШРИФТ 1: Для заголовков */
.заголовочный {
    border-left-color: #3498db;
    background-color: #ebf5fb;
```

```
}

.zаголовочный .название-шрифта {
    font-family: 'Playfair Display', serif;
    color: #2980b9;
    border-bottom-color: #3498db;
}

.zаголовочный .текст {
    font-family: 'Playfair Display', serif;
}

/* ШРИФТ 2: Для основного текста */
.основной {
    border-left-color: #2ecc71;
    background-color: #eafaf1;
}

.основной .название-шрифта {
    font-family: 'Lora', serif;
    color: #27ae60;
    border-bottom-color: #2ecc71;
}

.основной .текст {
    font-family: 'Lora', serif;
}

/* ШРИФТ 3: Для современного дизайна */
.современный {
    border-left-color: #9b59b6;
    background-color: #f4ecf7;
}

.современный .название-шрифта {
    font-family: 'Montserrat', sans-serif;
    color: #8e44ad;
    border-bottom-color: #9b59b6;
}

.современный .текст {
    font-family: 'Montserrat', sans-serif;
}
```

```
/* ШРИФТ 4: Для рукописных заметок */
.рукописный {
    border-left-color: #e74c3c;
    background-color: #fdedec;
}

.рукописный .название-шрифта {
    font-family: 'Caveat', cursive;
    color: #c0392b;
    border-bottom-color: #e74c3c;
}

.рукописный .текст {
    font-family: 'Caveat', cursive;
}

/* ШРИФТ 5: Для интерфейсов */
.интерфейсный {
    border-left-color: #f39c12;
    background-color: #fef9e7;
}

.интерфейсный .название-шрифта {
    font-family: 'Open Sans', sans-serif;
    color: #d35400;
    border-bottom-color: #f39c12;
}

.интерфейсный .текст {
    font-family: 'Open Sans', sans-serif;
}

/* Панель управления */
.панель-управления {
    background: #2c3e50;
    color: white;
    padding: 25px;
    border-radius: 15px;
    margin-top: 40px;
}

.кнопка {
    background: #3498db;
    color: white;
```

```
border: none;
padding: 12px 24px;
margin: 5px;
border-radius: 6px;
cursor: pointer;
font-size: 16px;
transition: background 0.3s;
}

.кнопка:hover {
background: #2980b9;
}

select {
padding: 10px;
font-size: 16px;
margin: 10px;
border-radius: 5px;
border: 2px solid #3498db;
width: 250px;
}

.предпросмотр {
background: white;
color: #333;
padding: 20px;
border-radius: 10px;
margin-top: 20px;
font-size: 20px;
min-height: 100px;
}

</style>
</head>
<body>
<div class="контейнер">
<h1> Моя шрифтовая книга</h1>
<p style="text-align: center; color: #666; margin-bottom: 40px;">
Коллекция красивых шрифтов для разных целей. Наведите на карточку для
эфекта!
</p>

<!-- Шрифт для заголовков -->
<div class="шрифт-карточка заголовочный">
<div class="название-шрифта">Playfair Display</div>
```

```
<div class="описание">Элегантный шрифт с засечками. Идеален для заголовков, приглашений, свадебных сайтов.</div>
<div class="примеры">
    <div class="пример">
        <div class="текст вес-400">
            Обычное начертание<br>
            «В начале было Слово»
        </div>
    </div>
    <div class="пример">
        <div class="текст вес-700">
            Жирное начертание<br>
            «И Слово было у Бога»
        </div>
    </div>
    <div class="пример">
        <div class="текст курсив">
            Курсив<br>
            «И Слово было Бог»
        </div>
    </div>
</div>
```

```
<!-- Шрифт для основного текста -->
<div class="шрифт-карточка основной">
    <div class="название-шрифта">Lora</div>
    <div class="описание">Уравновешенный шрифт с засечками. Отлично подходит для длинного текста, блогов, статей.</div>
    <div class="примеры">
        <div class="пример">
            <div class="текст вес-400">
                Ясный осенний вечер. Воздух прозрачен и свеж, и кажется, будто мир замер в ожидании чего-то важного.
            </div>
        </div>
    </div>
</div>
```

```
<!-- Шрифт современный -->
<div class="шрифт-карточка современный">
    <div class="название-шрифта">Montserrat</div>
    <div class="описание">Современный геометрический шрифт без засечек. Для интерфейсов, логотипов, мобильных приложений.</div>
```

```
<div class="примеры">
    <div class="пример">
        <div class="текст вес-300">
            Лёгкий вариант<br>
            Innovation happens
        </div>
    </div>
    <div class="пример">
        <div class="текст вес-400">
            Обычный вариант<br>
            when boundaries
        </div>
    </div>
    <div class="пример">
        <div class="текст вес-700">
            Жирный вариант<br>
            are crossed
        </div>
    </div>
</div>

<!-- Шрифт рукописный -->
<div class="шрифт-карточка рукописный">
    <div class="название-шрифта">Caveat</div>
    <div class="описание">Неформальный рукописный шрифт. Для открыток, заметок,
личных писем.</div>
    <div class="примеры">
        <div class="пример">
            <div class="текст вес-400">
                Дорогая бабушка!<br>
                Пишу тебе это письмо, чтобы сказать, как сильно я скучаю...
            </div>
        </div>
    </div>
</div>

<!-- Шрифт интерфейсный -->
<div class="шрифт-карточка интерфейсный">
    <div class="название-шрифта">Open Sans</div>
    <div class="описание">Нейтральный, читаемый шрифт без засечек. Один из самых
популярных для веба.</div>
    <div class="примеры">
        <div class="пример">
```

```
<div class="текст вес-300">
    User experience design is the process of enhancing user satisfaction
</div>
</div>
<div class="пример">
    <div class="текст вес-400">
        by improving the usability, accessibility, and pleasure provided
    </div>
</div>
<div class="пример">
    <div class="текст вес-700">
        in the interaction between the user and the product.
    </div>
</div>
</div>

<!-- Панель управления -->
<div class="панель-управления">
    <h2 style="color: white; margin-top: 0;">Песочница: создай свою комбинацию</h2>

    <div>
        <label for="выборШрифта">Выбери шрифт: </label>
        <select id="выборШрифта" onchange="обновитьПредпросмотр()">
            <option value="Playfair Display, serif">Playfair Display (заголовки)</option>
            <option value="Lora, serif">Lora (основной текст)</option>
            <option value="Montserrat, sans-serif">Montserrat (современный)</option>
            <option value="Caveat, cursive">Caveat (рукописный)</option>
            <option value="Open Sans, sans-serif" selected>Open Sans
                (интерфейсный)</option>
            <option value="Georgia, serif">Georgia (безопасный)</option>
            <option value="Arial, sans-serif">Arial (стандартный)</option>
            <option value="Courier New, monospace">Courier New
                (моноширинный)</option>
        </select>

        <label for="размерШрифта" style="margin-left: 20px;">Размер: </label>
        <select id="размерШрифта" onchange="обновитьПредпросмотр()">
            <option value="16px">Маленький (16px)</option>
            <option value="20px" selected>Средний (20px)</option>
            <option value="24px">Крупный (24px)</option>
            <option value="32px">Очень крупный (32px)</option>
        </select>
    </div>
```

```
<div style="margin-top: 15px;">
    <button class="кнопка" onclick="добавитьКурсив()">Курсив</button>
    <button class="кнопка" onclick="добавитьЖирный()">Жирный</button>
    <button class="кнопка"
        onclick="добавитьПодчеркивание()">Подчёркивание</button>
    <button class="кнопка" onclick="сброситьСтили()">Сбросить</button>
</div>

<div class="предпросмотр" id="предпросмотр">
    Введите свой текст здесь и увидите, как он будет выглядеть с выбранным
шрифтом!
</div>

<div style="margin-top: 15px;">
    <button class="кнопка" onclick="скопироватьКод()">📋 Скопировать CSS-
код</button>
    <button class="кнопка" onclick="случайныйШрифт()">🎲 Случайный
шрифт</button>
</div>
</div>

<script>
    function обновитьПредпросмотр() {
        const шрифт = document.getElementById('выборШрифта').value;
        const размер = document.getElementById('размерШрифта').value;
        const предпросмотр = document.getElementById('предпросмотр');

        предпросмотр.style.fontFamily = шрифт;
        предпросмотр.style.fontSize = размер;
        предпросмотр.style.fontWeight = 'normal';
        предпросмотр.style.fontStyle = 'normal';
        предпросмотр.style.textDecoration = 'none';
    }

    function добавитьКурсив() {
        const предпросмотр = document.getElementById('предпросмотр');
        предпросмотр.style.fontStyle =
            предпросмотр.style.fontStyle === 'italic' ? 'normal' : 'italic';
    }

    function добавитьЖирный() {
        const предпросмотр = document.getElementById('предпросмотр');
```

```
предпросмотр.style.fontWeight =
    предпросмотр.style.fontWeight === 'bold' ? 'normal' : 'bold';
}

function добавитьПодчеркивание() {
    const предпросмотр = document.getElementById('предпросмотр');
    предпросмотр.style.textDecoration =
        предпросмотр.style.textDecoration === 'underline' ? 'none' : 'underline';
}

function сброситьСтили() {
    const предпросмотр = document.getElementById('предпросмотр');
    предпросмотр.style.fontWeight = 'normal';
    предпросмотр.style.fontSize = 'normal';
    предпросмотр.style.textDecoration = 'none';
    обновитьПредпросмотр();
}

function скопироватьКод() {
    const шрифт = document.getElementById('выборШрифта').value;
    const размер = document.getElementById('размерШрифта').value;
    const код = `font-family: ${шрифт};\nfont-size: ${размер};`;

    navigator.clipboard.writeText(код)
        .then(() => alert('CSS-код скопирован в буфер обмена!'))
        .catch(err => alert('Не удалось скопировать код: ' + err));
}

function случайныйШрифт() {
    const шрифты =
        [
            "Playfair Display", serif",
            "Lora", serif",
            "Montserrat", sans-serif",
            "Caveat", cursive",
            "Open Sans", sans-serif",
            "Georgia, serif",
            "Arial, sans-serif",
            "Courier New", monospace",
            "Times New Roman", serif",
            "Verdana, sans-serif"
        ];
}

const случайныйИндекс = Math.floor(Math.random() * шрифты.length);
document.getElementById('выборШрифта').value = шрифты[случайныйИндекс];
```

```
    обновитьПредпросмотр();
}

// Позволяем редактировать текст предпросмотра
document.getElementById('предпросмотр').addEventListener('click', function() {
    const старыйТекст = this.textContent;
    const новыйТекст = prompt('Введите свой текст для предпросмотра:', старыйТекст);
    if (новыйТекст !== null) {
        this.textContent = новыйТекст;
    }
});

// Инициализация
обновитьПредпросмотр();
</script>
</body>
</html>
```

## Моя шрифтовая книга

Коллекция красивых шрифтов для разных целей. Наведите на карточку для эффекта!

### Playfair Display

Элегантный шрифт с ласочками. Идеален для заголовков, проклятий, свадебных сайтов.

Обычные начертания  
«И начало было Слово»

Жирные начертания  
«И Слово было у Бога»

Юрсив  
«И Слово было Бог»

### Lora

Удлиножиженный шрифт с ласочками. Отлично подходит для длинного текста, блогов, статей.

Вечный осенний вечер. Воздух прохладен и свеж, и кажется, будто мир замер в ожидании чего-то важного.

### Montserrat

Современный геометрический шрифт без ласочек. Для интерфейсов, логотипов, мобильных приложений.

Полный гарнитут  
Innovation happens

Обычный вариант  
when boundaries

Жирный вариант  
are crossed

### Caveat

Неформальный рукописный шрифт. Для открыток, заметок, личных писем.

Дорогой Роберт!  
Почуял запах чистого мыла, это лучше в горите...

### Open Sans

Нейтральный, чистый шрифт без ласочек. Одна из самых популярных для веба.

User experience design is the process of enhancing user satisfaction

by improving the usability, accessibility, and pleasure provided

In the interaction between the user and the product.

### Песочница: создай свою комбинацию

Выбери шрифт: Open Sans (интерфейс)

Размер: Средний (20px)

Курсив

Жирный

Подчеркивание

Образцы

Введите свой текст здесь и увидите, как он будет выглядеть с выбранным шрифтом!

Скопировать CSS-код

Случайный шрифт

---

## 6. Где искать красивые шрифты?

### Бесплатные источники:

1. [Google Fonts](#) — самый популярный, есть фильтры
2. [Font Squirrel](#) — качественные бесплатные шрифты
3. [DaFont](#) — огромная коллекция, много тематических
4. [1001 Fonts](#) — хороший поиск по категориям

### Поиск по настроению:

- **Элегантные:** Playfair Display, Garamond, Cormorant
- **Современные:** Montserrat, Raleway, Poppins
- **Дружелюбные:** Quicksand, Nunito, Comfortaa
- **Рукописные:** Caveat, Pacifico, Sacramento
- **Технические:** Roboto Mono, Source Code Pro
- **Русские:** Yandex Sans, PT Sans, Roboto (поддерживает кириллицу)

### Как проверить, есть ли кириллица:

На Google Fonts есть фильтр «Language» → выбирайте «Cyrillic» или «Cyrillic Extended».

---

## 7. Практические советы

### Правило 2-3 шрифтов:

css

/\* Хороший пример \*/

h1, h2, h3 {

```
font-family: 'Playfair Display', serif; /* Заголовки */  
}  
  
body {  
    font-family: 'Open Sans', sans-serif; /* Основной текст */  
}  
  
blockquote {  
    font-family: 'Caveat', cursive; /* Цитаты */  
}
```

## Универсальные комбинации:

1. **Playfair Display + Open Sans** (классика + современность)
2. **Montserrat + Merriweather** (строгий + уютный)
3. **Roboto + Roboto Slab** (гармоничное семейство)
4. **Lora + Source Sans Pro** (элегантность + простота)

## Что проверять:

1. Читается ли текст при размере 16px?
  2. Достаточно ли контраста между обычным и жирным?
  3. Хорошо ли сочетаются шрифты между собой?
  4. Не «кричат» ли они друг на друга?
- 

## 8. Домашнее задание

### Задание 1: «Моё резюме»

Создайте страницу-резюме с тщательно подобранными шрифтами:

- Имя и должность — элегантный шрифт
- Основной текст — читаемый sans-serif
- Контакты — моноширинный для адреса email

### Задание 2: «Шрифтовой контраст»

Создайте 3 колонки с одинаковым текстом, но разными комбинациями:

1. Serif заголовок + Sans-serif текст
2. Sans-serif заголовок + Serif текст

3. Один шрифт для всего, но разной жирности

### **Задание 3: «Настройте шрифтом»**

Напишите один абзац о радостном событии и оформите его тремя разными шрифтами:

1. Весёлый (Comic Sans или подобный)
2. Элегантный (с засечками)
3. Строгий (геометрический sans-serif)

### **Задание 4: «Создай палитру шрифтов»**

Сходите на Google Fonts и соберите свою коллекцию из 5 шрифтов:

1. Для заголовков блога
2. Для основного текста
3. Для цитат
4. Для подписей под фото
5. Для навигационного меню

**Бонус:** Найдите красивый РУССКИЙ рукописный шрифт (не все поддерживают кириллицу!)

---

## **9. Чего избегать?**

1. **Использовать слишком много шрифтов** — максимум 3 на страницу
2. **Декоративные шрифты для основного текста** — они плохо читаются
3. **Comic Sans для серьёзных текстов** — он имеет репутацию «несерьёзного»
4. **Papyrus, если вы не делаете сайт о Древнем Египте**
5. **Курсив для длинных текстов** — глаза быстро устают
6. **Слишком мелкий размер** — минимум 16px для основного текста

**Запомните:** Шрифт — это голос вашего текста. Он может говорить шёпотом, уверенно декламировать или весело кричать. Выбирайте голос, который подходит вашей аудитории и вашей цели!

Теперь вы — настоящий знаток шрифтов! Вы можете не только использовать стандартные, но и находить, подключать и правильно сочетать красивые

шрифты из интернета. В следующей главе мы научимся управлять размером букв — это будет просто, ведь вы уже освоили самое сложное! 🍪

## ■ **Размер букв (font-size): Почему px, em и % — как сантиметры и дюймы в шитье.**

Дорогой друг, представьте, что вы шьёте платье. Вы можете использовать сантиметры (точные измерения), дюймы (другая система) или сказать «в два раза шире, чем рукав» (относительные измерения). Так же и с размером шрифта в CSS! Давайте разберёмся, чем отличаются эти единицы измерения и когда какую использовать.

### **1. Основные единицы измерения — три «линейки»**

#### **A. Пиксели (px) — «Сантиметровая лента»**

**Пиксель** — это самая маленькая точка на экране. Когда вы говорите font-size: 16px;, вы говорите: «Сделай буквы высотой ровно 16 точек».

css

```
h1 {  
    font-size: 32px; /* Абсолютно 32 пикселя */  
}
```

**Как это выглядит:**

- 10px — очень мелко (как текст в сносках)
- 14px — мелко (как текст в газете)
- 16px — стандартный размер для веба
- 20px — крупно (подзаголовки)
- 32px — очень крупно (главные заголовки)
- 48px — огромно (баннеры)

**Преимущества px:**

- **Абсолютная точность** — всегда один и тот же размер
- **Предсказуемость** — вы точно знаете, что получите
- **Простота** — легко понять и использовать

**Недостатки px:**

- **Не адаптивно** — на маленьком экране 32px могут занимать пол-экрана

- Пользователь не может увеличить в некоторых старых браузерах
- Жёсткая привязка к конкретному размеру

#### Когда использовать px:

- Для элементов, размер которых должен быть **точно фиксированным**
  - Для границ, тени, мелких декоративных элементов
  - Когда вам нужен **полный контроль**
- 

## В. Проценты (%) — «Процент от размера родителя»

**Проценты** — это относительная единица. `font-size: 150%`; означает «на 50% больше, чем у родительского элемента».

css

```
body {  
    font-size: 16px; /* Базовый размер */  
}  
  
h1 {  
    font-size: 200%; /* 200% от 16px = 32px */  
}  
  
p {  
    font-size: 100%; /* 100% от 16px = 16px */  
}  
  
small {  
    font-size: 75%; /* 75% от 16px = 12px */  
}
```

#### Как это работает:

text

Родитель: 20px  
→ 50% = 10px  
→ 100% = 20px (тот же размер)  
→ 150% = 30px  
→ 200% = 40px

#### Преимущества %:

- **Относительность** — масштабируется вместе с родителем
- **Наследование** — легко создавать иерархию размеров
- **Гибкость** — меняешь базовый размер, все остальное подстраивается

#### Недостатки %:

- **Зависимость от родителя** — нужно знать, какой у родителя размер
- **Накопление** — если много вложенных элементов с процентами, может получиться неожиданный результат

#### Когда использовать %:

- Для создания **масштабируемых интерфейсов**
  - Когда нужно сохранить **пропорции** между элементами
  - Для **адаптивного дизайна** (вместе с медиа-запросами)
- 

## С. Единицы em — «Относительно текущего шрифта»

**1em** — это размер шрифта **текущего элемента** (или его родителя, если у текущего не задан).

css

```
body {  
    font-size: 16px; /* Базовый размер */  
}  
  
h1 {  
    font-size: 2em; /*  $2 \times 16px = 32px$  */  
}  
  
p {  
    font-size: 1em; /*  $1 \times 16px = 16px$  */  
}  
  
.крупный-текст {  
    font-size: 1.5em; /*  $1.5 \times 16px = 24px$  */  
}
```

**Особенность em:** Если применить **em** к вложенным элементам, они умножаются друг на друга!

css

```
div {  
    font-size: 20px;  
}  
  
div p {  
    font-size: 0.8em; /*  $0.8 \times 20px = 16px$  */  
}  
  
div p span {  
    font-size: 1.2em; /*  $1.2 \times 16px = 19.2px$  (а не  $1.2 \times 20px!$ ) */  
}
```

### Преимущества em:

- **Гибкость** — легко менять масштаб всего блока
- **Пропорциональность** — сохраняются отношения между элементами
- **Доступность** — пользователи могут увеличивать базовый размер

### Недостатки em:

- **Накопление** — как в примере выше, сложно предсказать конечный размер
- **Сложность расчётов** — нужно постоянно помнить о контексте

### Когда использовать em:

- Для **относительных отступов и размеров** (margin, padding)
  - Для **масштабируемых компонентов** (кнопок, карточек)
  - Когда нужно, чтобы **всё масштабировалось пропорционально**
- 

## D. Единицы rem — «Относительно корневого шрифта»

**rem** (root em) — всегда относительно размера шрифта элемента `<html>` (корневого).

css

```
html {  
    font-size: 16px; /* Это база для всех rem */  
}  
  
h1 {
```

```
font-size: 2rem; /* Всегда  $2 \times 16px = 32px$  */  
}  
  
p {  
    font-size: 1rem; /* Всегда  $1 \times 16px = 16px$  */  
}  
  
.контейнер {  
    font-size: 20px; /* Не влияет на rem внутри! */  
}  
  
.контейнер p {  
    font-size: 1.5rem; /* Все равно  $1.5 \times 16px = 24px$  */  
}
```

### Преимущества rem:

- **Предсказуемость** — не зависит от родительских элементов
- **Нет накопления** — всегда от корневого размера
- **Простота расчётов** — одна точка отсчёта
- **Идеально для адаптивности** — меняешь размер в html, всё масштабируется

### Недостатки rem:

- Меньше гибкости для изолированных компонентов
- Не подходит, если нужно масштабировать только часть страницы

### Когда использовать rem:

- **Для всего сайта** — основной способ задания размеров
  - **Для типографики** — заголовки, абзацы
  - **Для адаптивного дизайна** — главный инструмент
- 

## 2. Наглядное сравнение всех единиц

Создайте файл сравнение-размеров.html:

```
html  
<!DOCTYPE html>  
<html lang="ru">  
<head>
```

```
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Сравнение px, em, rem, %</title>
<style>
    /* Базовые настройки */
    :root {
        font-size: 16px; /* Базовый размер для rem */
    }

    body {
        font-family: 'Arial', sans-serif;
        line-height: 1.6;
        color: #333;
        padding: 30px;
        max-width: 900px;
        margin: 0 auto;
        background-color: #f9f9f9;
    }

    h1 {
        text-align: center;
        color: #2c3e50;
        margin-bottom: 40px;
        padding-bottom: 15px;
        border-bottom: 3px solid #3498db;
    }

    /* Контейнер для примеров */
    .линейка {
        background: white;
        padding: 25px;
        margin: 25px 0;
        border-radius: 15px;
        box-shadow: 0 5px 15px rgba(0,0,0,0.08);
        border-left: 8px solid;
    }

    .линейка h3 {
        margin-top: 0;
        padding-bottom: 10px;
        border-bottom: 2px solid;
    }

    /* Разные линейки по цветам */

```

```
.пиксели { border-left-color: #e74c3c; }
.пиксели h3 { border-bottom-color: #e74c3c; color: #e74c3c; }

.проценты { border-left-color: #2ecc71; }
.проценты h3 { border-bottom-color: #2ecc71; color: #2ecc71; }

.емы { border-left-color: #3498db; }
.емы h3 { border-bottom-color: #3498db; color: #3498db; }

.ремы { border-left-color: #9b59b6; }
.ремы h3 { border-bottom-color: #9b59b6; color: #9b59b6; }

/* Блоки с текстом разных размеров */
.размеры-ряд {
    display: flex;
    flex-wrap: wrap;
    gap: 15px;
    margin-top: 20px;
}

.текст-блок {
    padding: 15px;
    border-radius: 8px;
    background: #f8f9fa;
    flex: 1;
    min-width: 120px;
    text-align: center;
    border: 1px solid #ddd;
}

/* Контрольная панель */
.панель-контроля {
    background: #2c3e50;
    color: white;
    padding: 25px;
    border-radius: 15px;
    margin: 30px 0;
}

.ползунок {
    width: 100%;
    margin: 15px 0;
}
```

```
.кнопка {  
background: #3498db;  
color: white;  
border: none;  
padding: 12px 24px;  
margin: 5px;  
border-radius: 6px;  
cursor: pointer;  
font-size: 16px;  
}  
  
.
```

```
кнопка:hover {  
background: #2980b9;  
}
```

```
/* Пример с вложенностью для ем */  
.вложенный-пример {  
background: #fff8e1;  
padding: 20px;  
border-radius: 10px;  
margin: 20px 0;  
border: 2px dashed #ffb300;  
}
```

```
.уровень1 { font-size: 24px; padding: 15px; background: #e3f2fd; }  
.уровень2 { font-size: 0.8em; padding: 15px; background: #f3e5f5; margin-left: 20px; }  
.уровень3 { font-size: 0.8em; padding: 15px; background: #e8f5e9; margin-left: 40px; }
```

```
/* Информационные блоки */  
.инфо-блок {  
background: #e8f4fc;  
padding: 15px;  
border-radius: 8px;  
margin: 20px 0;  
border-left: 5px solid #3498db;  
}
```

```
.формула {  
font-family: 'Courier New', monospace;  
background: #2c3e50;  
color: white;  
padding: 10px;  
border-radius: 5px;  
margin: 10px 0;
```

```
    font-size: 14px;
}
</style>
</head>
<body>
<h1>🧵 Швейные линейки для веб-дизайна: px, %, em, rem</h1>

<div class="инфо-блок">
  <strong>Аналогия из шитья:</strong>
  <ul>
    <li><strong>px</strong> — сантиметровая лента (абсолютные значения: 10см, 20см)</li>
    <li><strong>%</strong> — «на 50% шире, чем спинка» (относительно другой детали)</li>
    <li><strong>em</strong> — «рукав в 1.5 раза шире манжеты» (относительно соседнего
      элемента)</li>
    <li><strong>rem</strong> — «все детали относительно выкройки-основы» (одна точка
      отсчёта)</li>
  </ul>
</div>

<!-- ПИКСЕЛИ -->
<div class="лнейка пиксели">
  <h3>1. Пиксели (px) — «Сантиметровая лента»</h3>
  <p>Абсолютные значения. 16px всегда будет 16 пикселями, независимо ни от чего.</p>

  <div class="формула">
    font-size: 16px; ← всегда 16 пикселей
  </div>

  <div class="размеры-ряд">
    <div class="текст-блок" style="font-size: 10px;">10px (мелкий)</div>
    <div class="текст-блок" style="font-size: 14px;">14px (газета)</div>
    <div class="текст-блок" style="font-size: 16px;">16px (стандарт)</div>
    <div class="текст-блок" style="font-size: 20px;">20px (крупный)</div>
    <div class="текст-блок" style="font-size: 32px;">32px (заголовок)</div>
  </div>

  <div class="инфо-блок">
    <strong>Плюсы:</strong> Точность, предсказуемость<br>
    <strong>Минусы:</strong> Не адаптивно, пользователь не может увеличить<br>
    <strong>Когда использовать:</strong> Для фиксированных размеров (иконки, границы)
  </div>
</div>

<!-- ПРОЦЕНТЫ -->
```

```
<div class="линейка проценты">
  <h3>2. Проценты (%) — «Процент от родителя»</h3>
  <p>Относительно размера шрифта родительского элемента.</p>
```

```
<div class="формула">
  Родитель: 20px → 150% = 30px (20 × 1.5)
</div>
```

```
<div style="font-size: 20px; background: #f0f8ff; padding: 15px; border-radius: 8px;">
  <strong>Родительский блок (font-size: 20px):</strong>
  <div style="font-size: 50%; margin-top: 10px;">Это 50% → 10px (20 × 0.5)</div>
  <div style="font-size: 100%;">Это 100% → 20px (тот же размер)</div>
  <div style="font-size: 150%;">Это 150% → 30px (20 × 1.5)</div>
  <div style="font-size: 200%;">Это 200% → 40px (20 × 2)</div>
</div>
```

```
<div class="инфо-блок">
  <strong>Плюсы:</strong> Относительность, наследование<br>
  <strong>Минусы:</strong> Зависимость от родителя, сложные расчёты<br>
  <strong>Когда использовать:</strong> Для масштабируемых компонентов
</div>
</div>
```

```
<!-- EM -->
<div class="линейка емы">
  <h3>3. Единицы em — «Относительно текущего шрифта»</h3>
  <p>1em = текущий размер шрифта элемента. Если не задан — берётся у родителя.</p>
```

```
<div class="формула">
  font-size: 1.5em; ← в 1.5 раза больше текущего размера
</div>
```

```
<!-- Пример с накоплением -->
<div class="вложенный-пример">
  <p><strong>Пример накопления em (осторожно!):</strong></p>
  <div class="уровень1">
    Уровень 1: font-size: 24px
    <div class="уровень2">
      Уровень 2: font-size: 0.8em (24px × 0.8 = 19.2px)
      <div class="уровень3">
        Уровень 3: font-size: 0.8em (19.2px × 0.8 = 15.36px)
      </div>
    </div>
  </div>
</div>
```

```
<p style="color: #d35400; margin-top: 10px;">
  ⚠️ Обратите внимание: каждый следующий уровень умножается на предыдущий размер!
</p>
</div>

<div class="инфо-блок">
  <strong>Плюсы:</strong> Гибкость, пропорциональность<br>
  <strong>Минусы:</strong> Накопление, сложные расчёты<br>
  <strong>Когда использовать:</strong> Для отступов, масштабируемых компонентов
</div>
</div>

<!-- REM -->
<div class="лайнейка ремы">
  <h3>4. Единицы rem — «Относительно корневого шрифта»</h3>
  <p>1rem = размер шрифта элемента &lt;html&gt; (по умолчанию 16px). Не зависит от
родителей!</p>

<div class="формула">
  html { font-size: 16px; }<br>
  h1 { font-size: 2rem; } ← всегда 32px (16 × 2)
</div>

<div style="background: #f0f8ff; padding: 15px; border-radius: 8px;">
  <strong>Корневой размер: <span id="корневойРазмер">16px</span></strong>
  <div class="размеры-ряд">
    <div class="текст-блок" style="font-size: 0.5rem;">0.5rem → <span
id="rem05">8px</span></div>
    <div class="текст-блок" style="font-size: 1rem;">1rem → <span id="rem1">16px</span></div>
    <div class="текст-блок" style="font-size: 1.5rem;">1.5rem → <span
id="rem15">24px</span></div>
    <div class="текст-блок" style="font-size: 2rem;">2rem → <span id="rem2">32px</span></div>
    <div class="текст-блок" style="font-size: 3rem;">3rem → <span id="rem3">48px</span></div>
  </div>
</div>

<!-- Пример без накопления -->
<div style="font-size: 30px; background: #e8f5e9; padding: 15px; border-radius: 8px; margin-top:
20px;">
  <strong>Родитель с font-size: 30px</strong>
  <div style="font-size: 1rem; background: #c8e6c9; padding: 10px; margin-top: 10px;">
    Это 1rem → все равно 16px (не зависит от родителя!)
  </div>
</div>
```

```
<div class="инфо-блок">
    <strong>Плюсы:</strong> Предсказуемость, нет накопления, адаптивность<br>
    <strong>Минусы:</strong> Меньше гибкости для изолированных блоков<br>
    <strong>Когда использовать:</strong> Для всей типографики сайта
</div>
</div>
```

```
<!-- ПАНЕЛЬ УПРАВЛЕНИЯ -->
<div class="панель-контроля">
    <h3 style="color: white; margin-top: 0;">Лаборатория: Изменяем базовый размер</h3>
```

```
<label for="ползунокRem" style="display: block; margin: 15px 0;">
    Измените корневой размер шрифта (влияет на rem):
</label>
<input type="range" id="ползунокRem" min="8" max="32" value="16" class="ползунок">
<div style="display: flex; justify-content: space-between;">
    <span>8px (мелко)</span>
    <span id="текущееЗначение">16px</span>
    <span>32px (крупно)</span>
</div>
```

```
<div style="margin-top: 20px;">
    <strong>Быстрые пресеты:</strong>
    <button class="кнопка" onclick="установитьРазмер(12)">  Мобильный (12px)</button>
    <button class="кнопка" onclick="установитьРазмер(16)">  Десктоп (16px)</button>
    <button class="кнопка" onclick="установитьРазмер(20)">  Крупный (20px)</button>
    <button class="кнопка" onclick="установитьРазмер(24)">  Презентация (24px)</button>
</div>
```

```
<div style="margin-top: 20px; background: rgba(255,255,255,0.1); padding: 15px; border-radius: 8px;">
    <strong>Что происходит:</strong> Меняется размер в <code>html { font-size: ... }</code><br>
    Все элементы с <strong>rem</strong> мгновенно масштабируются!<br>
    Элементы с <strong>px</strong> не меняются (абсолютные значения)<br>
    Элементы с <strong>%</strong> и <strong>em</strong> меняются, если их родители
```

используют rem

```
</div>
</div>
```

```
<!-- ПРАКТИЧЕСКИЕ СОВЕТЫ -->
<div style="background: #fff8e1; padding: 25px; border-radius: 15px; margin: 30px 0; border-left: 8px solid #ffb300;">
    <h3 style="color: #ff8f00; margin-top: 0;">  Практические советы по выбору единиц</h3>
```

```
<div style="display: grid; grid-template-columns: repeat(auto-fit, minmax(250px, 1fr)); gap: 20px; margin-top: 20px;">
  <div style="background: white; padding: 15px; border-radius: 8px;">
    <strong style="color: #e74c3c;">Используйте px:</strong>
    <ul>
      <li>Для границ (border: 1px)</li>
      <li>Для теней (box-shadow)</li>
      <li>Для маленьких фиксированных элементов</li>
      <li>Когда нужна абсолютная точность</li>
    </ul>
  </div>

  <div style="background: white; padding: 15px; border-radius: 8px;">
    <strong style="color: #2ecc71;">Используйте %:</strong>
    <ul>
      <li>Для ширины и высоты блоков</li>
      <li>Вместе с media queries</li>
      <li>Для изображений (max-width: 100%)</li>
      <li>В layout (ширина колонок)</li>
    </ul>
  </div>

  <div style="background: white; padding: 15px; border-radius: 8px;">
    <strong style="color: #3498db;">Используйте em:</strong>
    <ul>
      <li>Для padding и margin текстовых блоков</li>
      <li>Внутри изолированных компонентов</li>
      <li>Когда нужна пропорциональность внутри блока</li>
      <li>Для line-height (1.5em)</li>
    </ul>
  </div>

  <div style="background: white; padding: 15px; border-radius: 8px;">
    <strong style="color: #9b59b6;">Используйте rem:</strong>
    <ul>
      <li>Для всей типографики (font-size)</li>
      <li>Для адаптивного дизайна</li>
      <li>Для основных размеров интерфейса</li>
      <li>Как основную единицу измерения</li>
    </ul>
  </div>
</div>
```

```
<div class="формула" style="margin-top: 20px; background: #2c3e50; color: white;">
     <strong>Золотое правило:</strong><br>
    font-size: rem; ← для текста<br>
    padding/margin: em; ← для отступов вокруг текста<br>
    width/height: % или px; ← для размеров блоков<br>
    media queries: rem или em; ← для адаптивности
</div>
</div>
```

<!-- ДОМАШНЕЕ ЗАДАНИЕ -->

```
<div style="background: #e8f4fc; padding: 25px; border-radius: 15px; margin: 30px 0; border: 2px dashed #3498db;">
    <h3 style="color: #2980b9; margin-top: 0;"> Домашнее задание</h3>
```

<p><strong>Задание 1:</strong> Создайте страницу, где:</p>

<ul>

<li>Заголовок h1 = 2.5rem</li>

<li>Заголовок h2 = 2rem</li>

<li>Основной текст = 1.2rem</li>

<li>Мелкий текст = 0.875rem</li>

<li>Измените корневой размер с 16px на 18px — всё должно пропорционально  
увеличиться</li>

</ul>

<p><strong>Задание 2:</strong> Создайте кнопку, где:</p>

<ul>

<li>Текст кнопки = 1rem</li>

<li>Padding = 1em (чтобы был пропорционален тексту)</li>

<li>Border-radius = 0.5em</li>

<li>Попробуйте изменить размер текста — padding и border-radius подстроятся  
автоматически!</li>

</ul>

<p><strong>Задание 3:</strong> Поэкспериментируйте с ползунком выше. Посмотрите, как  
меняются:</p>

<ul>

<li>Элементы с rem (меняются сразу)</li>

<li>Элементы с px (не меняются)</li>

<li>Элементы с em в примере с вложенностью (меняются сложно)</li>

</ul>

</div>

<script>

// Обновляем значения при изменении ползунка

```
const ползунок = document.getElementById('ползунокRem');
const текущееЗначение = document.getElementById('текущееЗначение');

function обновитьРазмеры() {
    const размер = ползунок.value;

    // Меняем корневой размер
    document.documentElement.style.fontSize = размер + 'px';

    // Обновляем отображение
    текущееЗначение.textContent = размер + 'px';
    document.getElementById('корневойРазмер').textContent = размер + 'px';

    // Пересчитываем значения для rem-блоков
    document.getElementById('rem05').textContent = Math.round(размер * 0.5) + 'px';
    document.getElementById('rem1').textContent = Math.round(размер * 1) + 'px';
    document.getElementById('rem15').textContent = Math.round(размер * 1.5) + 'px';
    document.getElementById('rem2').textContent = Math.round(размер * 2) + 'px';
    document.getElementById('rem3').textContent = Math.round(размер * 3) + 'px';
}

function установитьРазмер(размер) {
    ползунок.value = размер;
    обновитьРазмеры();
}

// Инициализация
ползунок.addEventListener('input', обновитьРазмеры);
обновитьРазмеры(); // Первоначальный расчёт
</script>
</body>
</html>
```



## Швейные линейки для веб-дизайна: px, %, em, rem

### Аналогия из шитья:

- **px** — сантиметровая лента (абсолютные значения: 10см, 20см)
- **%** — «на 50% шире, чем спинка» (относительно другой детали)
- **em** — «рукав в 1.5 раза шире манжеты» (относительно соседнего элемента)
- **rem** — «все детали относительно выкройки-основы» (одна точка отсчёта)

### 1. Пиксели (px) — «Сантиметровая лента»

Абсолютные значения. 16px всегда будет 16 пикселями, независимо ни от чего.

`font-size: 16px; ← всегда 16 пикселей`

10px (мелкий)

14px (газета)

16px (стандарт)

20px  
(крупный)

32px  
(заголовок)

**Плюсы:** Точность, предсказуемость

**Минусы:** Не адаптивно, пользователь не может увеличить

**Минусы:** Не адаптивно, пользователь не может увеличить

**Когда использовать:** Для фиксированных размеров (иконки, границы)

### 2. Проценты (%) — «Процент от родителя»

Относительно размера шрифта родительского элемента.

`Родитель: 20px → 150% = 30px (20 × 1.5)`

**Родительский блок (font-size: 20px):**

Это 50% → 10px ( $20 \times 0.5$ )

Это 100% → 20px (тот же размер)

Это 150% → 30px ( $20 \times 1.5$ )

**Это 200% → 40px ( $20 \times 2$ )**

**Плюсы:** Относительность, наследование

**Минусы:** Зависимость от родителя, сложные расчёты

**Когда использовать:** Для масштабируемых компонентов

### 3. Практическое применение: Создаём адаптивную типографику

**Правильный подход для современного сайта:**

css

```
/* 1. Устанавливаем базовый размер для rem */
html {
    font-size: 16px; /* По умолчанию */
}

/* 2. Для адаптивности меняем на разных экранах */
@media (max-width: 768px) {
    html {
        font-size: 14px; /* На мобильных чуть мельче */
    }
}

@media (min-width: 1200px) {
    html {
        font-size: 18px; /* На больших экранах крупнее */
    }
}

/* 3. Используем rem для типографики */
body {
    font-size: 1rem; /* = 16px на десктопе */
    line-height: 1.6;
}

h1 {
    font-size: 2.5rem; /* 40px на десктопе */
    margin-bottom: 1.5rem; /* 24px */
}

h2 {
    font-size: 2rem; /* 32px */
    margin-bottom: 1rem; /* 16px */
}

p {
    font-size: 1.125rem; /* 18px */
```

```

margin-bottom: 1rem;
}

/* 4. Используем em для элементов, связанных с текстом */
.button {
    font-size: 1rem;
    padding: 0.75em 1.5em; /* Пропорционально тексту */
    border-radius: 0.5em;
}

/* 5. Используем px для фиксированных элементов */
.container {
    max-width: 1200px; /* Фиксированная максимальная ширина */
    margin: 0 auto;
    padding: 0 15px; /* Фиксированные отступы */
}

```

## 4. Шпаргалка «Что выбрать?»

Задача	Лучшая единица	Пример	Почему
<b>Размер шрифта</b>	rem	font-size: 1.25rem;	Адаптивно, предсказуемо
<b>Отступы вокруг текста</b>	em	padding: 1em;	Пропорционально тексту
<b>Высота строки</b>	em	line-height: 1.6em;	Относительно размера шрифта
<b>Ширина блоков</b>	%	width: 50%;	Относительно родителя
<b>Максимальная ширина</b>	px или %	max-width: 1200px;	Фиксировано или относительно
<b>Границы, тени</b>	px	border: 1px solid;	Абсолютная точность
<b>Адаптивные breakpoints</b>	em или rem	@media (min-width: 48em)	Зависит от размера шрифта

## 5. Частые ошибки новичков

### 1. Смешивание единиц без понимания:

css

```
/* ПЛОХО: сложно понять итоговый размер */
.block {
    font-size: 14px;
    padding: 1.5em; /* 21px? Зависит от родителя! */
}
```

## 2. Использование px для всего:

css

```
/* ПЛОХО: не адаптивно, недоступно */
body { font-size: 14px; }
h1 { font-size: 28px; }
p { font-size: 14px; }
```

## 3. Забывают про line-height:

css

```
/* ПЛОХО: текст слипается */
p {
    font-size: 1.2rem;
    /* line-height не задан! */
}

/* ХОРОШО: */
p {
    font-size: 1.2rem;
    line-height: 1.6; /* или 1.6em */
}
```

# 6. Дополнительные единицы измерения

Есть и другие, но они используются реже:

- **vw/vh** — процент от ширины/высоты окна

css

```
h1 { font-size: 5vw; } /* 5% от ширины окна */
```

- **vmin/vmax** — процент от меньшей/большей стороны
- **ch** — ширина символа "0" текущего шрифта

- **ex** — высота символа "x" текущего шрифта

**Совет:** Начинайте с `rem` для типографики, `em` для связанных размеров, `%` для `layout`, `px` для мелких фиксированных элементов.

## 7. Итог: простое правило

**Для пенсионеров (и не только):**

1. **Хотите, чтобы всё масштабировалось вместе?** → Используйте `rem`
2. **Хотите, чтобы отступы были пропорциональны тексту?** → Используйте `em`
3. **Хотите точный, неизменный размер?** → Используйте `px`
4. **Хотите, чтобы размер зависел от родителя?** → Используйте `%`

**Проще всего:** Начните с `rem` для всего текста. Это безопасно, предсказуемо и адаптивно!

Теперь вы знаете все «линейки» веб-дизайна! Вы можете точно измерять, гибко масштабировать и создавать удобные для чтения интерфейсы. В следующей главе мы научимся делать текст жирным и наклонным — это проще простого!

## ■ **Жирность (font-weight) и наклон (font-style): Выделяем важное.**

Дорогой друг, представьте, что вы разговариваете с кем-то. Чтобы подчеркнуть важные слова, вы можете сказать их громче (это font-weight) или с особым выражением (это font-style). В тексте мы делаем то же самое! Давайте научимся управлять "громкостью" и "выражением" наших букв.

### **1. Жирность текста (font-weight) — «Как громко говорить»**

#### **Что такое font-weight?**

Это свойство определяет **насыщенность** или **толщину** шрифта. От тонкого, едва заметного, до жирного, бросающегося в глаза.

#### **Способы задания жирности:**

##### **A. Ключевые слова (самый простой способ)**

css

```
p {  
    font-weight: normal; /* Обычная толщина (400) */  
    font-weight: bold; /* Жирный (700) */  
    font-weight: lighter; /* Тоньше, чем у родителя */  
    font-weight: bolder; /* Жирнее, чем у родителя */  
}
```

##### **B. Числовые значения (более точный контроль)**

css

```
p {  
    font-weight: 100; /* Thin (очень тонкий) */  
    font-weight: 200; /* Extra Light */  
    font-weight: 300; /* Light (светлый) */
```

```
font-weight: 400; /* Normal (обычный) */  
font-weight: 500; /* Medium (средний) */  
font-weight: 600; /* Semi Bold (полужирный) */  
font-weight: 700; /* Bold (жирный) */  
font-weight: 800; /* Extra Bold */  
font-weight: 900; /* Black (очень жирный) */  
}
```

**Важно:** Не все шрифты имеют все эти варианты! Многие имеют только 400 (normal) и 700 (bold).

---

## 2. Наклон текста (`font-style`) — «С каким выражением говорить»

### Что такое `font-style`?

Это свойство определяет **наклон** символов шрифта.

### Основные значения:

css

```
p {  
    font-style: normal; /* Обычное начертание (без наклона) */  
    font-style: italic; /* Курсив (специальный наклонный вариант шрифта) */  
    font-style: oblique; /* Наклонный (просто наклонённая версия обычного шрифта) */  
}
```

### Разница между `italic` и `oblique`:

- **Italic** — это отдельный, специально созданный наклонный вариант шрифта (обычно более красивый)
- **Oblique** — просто наклон обычного шрифта (как будто наклонили в графическом редакторе)

На практике почти всегда используют `italic`, так как он выглядит лучше.

---

### 3. Наглядный пример всех вариантов

Создайте файл жирность-наклон.html:

html

```
<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Жирность и наклон текста</title>

    <!-- Подключаем шрифт с разными начертаниями -->
    <link rel="preconnect" href="https://fonts.googleapis.com">
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
    <link href="https://fonts.googleapis.com/css2?
family=Roboto:ital,wght@0,100;0,300;0,400;0,500;0,700;0,900;1,100;1,300;1,400;1,500;1,700;1,900&display=swap" rel="stylesheet">

<style>
    /* Общие настройки */
    * {
        box-sizing: border-box;
    }

    body {
        font-family: 'Roboto', sans-serif;
        line-height: 1.6;
        color: #333;
        background-color: #f8f9fa;
        padding: 30px;
        max-width: 1000px;
        margin: 0 auto;
    }

    h1 {
        text-align: center;
        color: #2c3e50;
        margin-bottom: 40px;
        padding-bottom: 15px;
        border-bottom: 3px solid #3498db;
    }
```

```
.раздел {  
    background: white;  
    padding: 30px;  
    margin: 30px 0;  
    border-radius: 15px;  
    box-shadow: 0 5px 20px rgba(0,0,0,0.08);  
}  
  
.раздел h2 {  
    color: #2c3e50;  
    margin-top: 0;  
    padding-bottom: 10px;  
    border-bottom: 2px solid;  
}  
  
/* Цвета для разных разделов */  
.жирность-раздел { border-left: 8px solid #e74c3c; }  
.жирность-раздел h2 { border-bottom-color: #e74c3c; }  
  
.наклон-раздел { border-left: 8px solid #3498db; }  
.наклон-раздел h2 { border-bottom-color: #3498db; }  
  
.комбинированный-раздел { border-left: 8px solid #2ecc71; }  
.комбинированный-раздел h2 { border-bottom-color: #2ecc71; }  
  
.практический-раздел { border-left: 8px solid #9b59b6; }  
.практический-раздел h2 { border-bottom-color: #9b59b6; }  
  
/* Сетка для демонстрации */  
.сетка-жирности {  
    display: grid;  
    grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));  
    gap: 20px;  
    margin-top: 25px;  
}  
  
.карточка-жирности {  
    padding: 20px;  
    border-radius: 10px;  
    background: #f8f9fa;  
    border: 1px solid #e9ecf;  
    transition: transform 0.3s;  
}
```

```
.карточка-жирности:hover {
    transform: translateY(-5px);
    box-shadow: 0 8px 15px rgba(0,0,0,0.1);
}

.значение {
    font-size: 18px;
    font-weight: bold;
    color: #2c3e50;
    margin-bottom: 10px;
    padding-bottom: 5px;
    border-bottom: 1px dashed #ccc;
}

.пример-текста {
    font-size: 20px;
    min-height: 60px;
    display: flex;
    align-items: center;
}

/* Примеры наклона */
.примеры-наклона {
    display: flex;
    flex-wrap: wrap;
    gap: 30px;
    margin-top: 25px;
}

.карточка-наклона {
    flex: 1;
    min-width: 250px;
    padding: 25px;
    border-radius: 10px;
    background: #f8f9fa;
}

/* Практические примеры */
.практический-пример {
    background: #fff;
    padding: 25px;
    border-radius: 10px;
    margin: 20px 0;
    border: 2px solid #e9ecf;
```

```
}

.хороший-пример { border-left: 6px solid #2ecc71; }

.плохой-пример { border-left: 6px solid #e74c3c; }

/* Интерактивная панель */
.лаборатория {
    background: #2c3e50;
    color: white;
    padding: 30px;
    border-radius: 15px;
    margin: 30px 0;
}

.управление {
    display: grid;
    grid-template-columns: repeat(auto-fit, minmax(200px, 1fr));
    gap: 20px;
    margin-top: 20px;
}

.кнопка-стиль {
    background: #3498db;
    color: white;
    border: none;
    padding: 15px;
    border-radius: 8px;
    cursor: pointer;
    font-size: 16px;
    transition: all 0.3s;
}

.кнопка-стиль:hover {
    background: #2980b9;
    transform: translateY(-2px);
}

.жирная { background: #e74c3c; }

.жирная:hover { background: #c0392b; }

.наклонная { background: #9b59b6; }

.наклонная:hover { background: #8e44ad; }

.комбинированная { background: #2ecc71; }
```

```
.комбинированная:hover { background: #27ae60; }

.предпросмотр {
    background: white;
    color: #333;
    padding: 30px;
    border-radius: 10px;
    margin-top: 25px;
    font-size: 24px;
    min-height: 120px;
    display: flex;
    align-items: center;
    justify-content: center;
    text-align: center;
}

/* Информационные блоки */
.инфо-блок {
    background: #e8f4fc;
    padding: 20px;
    border-radius: 10px;
    margin: 20px 0;
    border-left: 5px solid #3498db;
}

.предупреждение {
    background: #fef9e7;
    padding: 20px;
    border-radius: 10px;
    margin: 20px 0;
    border-left: 5px solid #f39c12;
    color: #7d6608;
}

/* Цитата для демонстрации */
.красивая-цитата {
    font-family: 'Georgia', serif;
    font-size: 24px;
    line-height: 1.5;
    color: #2c3e50;
    padding: 40px;
    border-radius: 15px;
    background: linear-gradient(135deg, #f5f7fa 0%, #c3cfe2 100%);
    margin: 30px 0;
}
```

```
position: relative;
}

.красивая-цитата::before {
    content: "";
    font-size: 100px;
    color: #3498db;
    position: absolute;
    top: -20px;
    left: 20px;
    opacity: 0.3;
}

.автор-цитаты {
    text-align: right;
    font-style: italic;
    color: #7f8c8d;
    margin-top: 20px;
    font-size: 18px;
}
}

</style>
</head>
<body>
<h1>🎨 Жирность (font-weight) и наклон (font-style): Искусство выделения</h1>

<!-- Раздел 1: Жирность -->
<div class="раздел жирность-раздел">
    <h2>1. Шкала жирности (font-weight)</h2>
    <p>От самого тонкого (100) до самого жирного (900). Не все шрифты поддерживают все значения.</p>

    <div class="сетка-жирности">
        <!-- Тонкие начертания -->
        <div class="карточка-жирности">
            <div class="значение">100 (Thin) — Очень тонкий</div>
            <div class="пример-текста" style="font-weight: 100;">
                Лёгкое дыхание весны
            </div>
        </div>

        <div class="карточка-жирности">
            <div class="значение">200 (Extra Light)</div>
            <div class="пример-текста" style="font-weight: 200;">
                Утренний туман над рекой
            </div>
        </div>
    </div>
</div>
```

```
</div>
</div>

<div class="карточка-жирности">
    <div class="значение">300 (Light) — Светлый</div>
    <div class="пример-текста" style="font-weight: 300;">
        Шёпот осеннего листопада
    </div>
</div>

<!-- Стандартные -->
<div class="карточка-жирности">
    <div class="значение">400 (Normal) — Обычный</div>
    <div class="пример-текста" style="font-weight: 400;">
        Обычный текст для чтения
    </div>
</div>

<div class="карточка-жирности">
    <div class="значение">500 (Medium) — Средний</div>
    <div class="пример-текста" style="font-weight: 500;">
        Важное, но не кричащее
    </div>
</div>

<!-- Жирные -->
<div class="карточка-жирности">
    <div class="значение">600 (Semi Bold) — Полужирный</div>
    <div class="пример-текста" style="font-weight: 600;">
        Обратите внимание!
    </div>
</div>

<div class="карточка-жирности">
    <div class="значение">700 (Bold) — Жирный</div>
    <div class="пример-текста" style="font-weight: 700;">
        ОЧЕНЬ ВАЖНО!
    </div>
</div>

<div class="карточка-жирности">
    <div class="значение">800 (Extra Bold)</div>
    <div class="пример-текста" style="font-weight: 800;">
        КРИТИЧЕСКИ ВАЖНО!!!
    </div>
</div>
```

```
</div>
</div>

<div class="карточка-жирности">
    <div class="значение">900 (Black) — Очень жирный</div>
    <div class="пример-текста" style="font-weight: 900;">
        НЕОБХОДИМО СРОЧНО!
    </div>
</div>
</div>

<div class="инфо-блок">
    <strong>💡 Практический совет:</strong> Для основного текста используйте 300-400, для заголовков — 600-700, для особого акцента — 800-900. Но не переусердствуйте — слишком жирный текст утомляет глаза.
</div>
</div>

<!-- Раздел 2: Наклон --&gt;
&lt;div class="раздел наклон-раздел"&gt;
    &lt;h2&gt;2. Варианты наклона (font-style)&lt;/h2&gt;
    &lt;p&gt;Курсив (italic) и наклонный (oblique) — два способа сделать текст наклонным.&lt;/p&gt;

    &lt;div class="примеры-наклона"&gt;
        &lt;div class="карточка-наклона"&gt;
            &lt;div class="значение"&gt;font-style: normal&lt;/div&gt;
            &lt;div class="пример-текста" style="font-style: normal; font-size: 22px;"&gt;
                Это обычный текст, без наклона. Используется для основного содержания.
            &lt;/div&gt;
        &lt;/div&gt;
    &lt;/div&gt;

    &lt;div class="карточка-наклона"&gt;
        &lt;div class="значение"&gt;font-style: italic&lt;/div&gt;
        &lt;div class="пример-текста" style="font-style: italic; font-size: 22px;"&gt;
            Это курсив — специальный наклонный вариант шрифта. Выглядит изящно.
        &lt;/div&gt;
    &lt;/div&gt;

    &lt;div class="карточка-наклона"&gt;
        &lt;div class="значение"&gt;font-style: oblique&lt;/div&gt;
        &lt;div class="пример-текста" style="font-style: oblique; font-size: 22px;"&gt;
            Это просто наклон обычного шрифта. Используется редко.
        &lt;/div&gt;
    &lt;/div&gt;
</pre>
```

```
</div>
</div>
</div>

<div class="инфо-блок">
    <strong>⌚ Когда использовать курсив:</strong>
    <ul>
        <li>Для выделения мыслей, цитат</li>
        <li>Для иностранных слов (например, <i>et cetera</i>)</li>
        <li>Для названий книг, фильмов, картин</li>
        <li>Для научных терминов</li>
        <li>Для подписей к иллюстрациям</li>
    </ul>
</div>

<div class="предупреждение">
    <strong>⚠ Внимание:</strong> Не используйте курсив для длинных текстов — он
хуже читается!
    Максимум 2-3 строки.
    </div>
</div>

<!-- Раздел 3: Комбинирование -->
<div class="раздел комбинированный-раздел">
    <h2>3. Комбинирование жирности и наклона</h2>
    <p>Можно одновременно использовать и жирность, и курсив для особого акцента.</p>

    <div class="красивая-цитата">
        <span style="font-weight: 300; font-style: italic;">Искусство жить</span>
        <span style="font-weight: 400; font-style: normal;">заключается не в том, чтобы
избегать бурь,</span>
        <span style="font-weight: 500; font-style: normal;">а в том, чтобы</span>
        <span style="font-weight: 700; font-style: italic;">танцевать под дождём.</span>
        <div class="автор-цитаты">— Неизвестный автор</div>
    </div>

    <div class="сетка-жирности" style="grid-template-columns: repeat(auto-fit, minmax(300px,
1fr));">
        <div class="карточка-жирности">
            <div class="значение">Light Italic (300 italic)</div>
            <div class="пример-текста" style="font-weight: 300; font-style: italic;">
                Нежное напоминание
            </div>
        </div>
    </div>
```

```
<div class="карточка-жирности">
    <div class="значение">Normal Italic (400 italic)</div>
    <div class="пример-текста" style="font-weight: 400; font-style: italic;">
        Обычный курсив для цитат
    </div>
</div>

<div class="карточка-жирности">
    <div class="значение">Bold Italic (700 italic)</div>
    <div class="пример-текста" style="font-weight: 700; font-style: italic;">
        Важная цитата!
    </div>
</div>
</div>
```

<!-- Раздел 4: Практические примеры -->

```
<div class="раздел практический-раздел">
    <h2>4. Практические примеры использования</h2>
```

```
<div class="практический-пример хороший-пример">
    <h3>✓ Хороший пример: Удобочитаемый текст</h3>
    <p style="font-weight: 300; font-size: 18px; line-height: 1.6;">
        Основной текст статьи или блога лучше делать <strong>лёгким (300)</strong>
    или <strong>нормальным (400)</strong>.
```

Это снижает нагрузку на глаза при длительном чтении.

```
</p>
<p style="font-weight: 600; font-size: 20px; margin-top: 15px;">
    Заголовки должны быть <strong>полужирными (600)</strong> или
<strong>жирными (700)</strong>, чтобы сразу привлекать внимание.
```

```
</p>
<p style="font-style: italic; color: #666; margin-top: 15px;">
    <strong>Цитаты и важные мысли</strong> можно выделять курсивом, но не
```

слишком длинные.

```
</p>
</div>
```

```
<div class="практический-пример плохой-пример">
    <h3>✗ Плохой пример: Перегруженный текст</h3>
    <p style="font-weight: 900; font-size: 18px;">
```

ЕСЛИ ВЕСЬ ТЕКСТ СДЕЛАТЬ ОЧЕНЬ ЖИРНЫМ (900), ТО ЧИТАТЬ ЕГО БУДЕТ ОЧЕНЬ  
ТЯЖЕЛО.

ГЛАЗА БЫСТРО УСТАЮТ ОТ ВЫСОКОГО КОНТРАСТА.

```
</p>
<p style="font-weight: 700; font-style: italic; margin-top: 15px;">
    А ЕЩЁ И ДОБАВИТЬ КУРСИВ, ТО ВООБЩЕ СТАНОВИТСЯ НЕВОЗМОЖНО
РАЗОБРАТЬ,
    <span style="font-weight: 900; text-decoration: underline;">ЧТО ТУТ
НАПИСАНО</span>.
</p>
</div>
```

```
<div class="инфо-блок">
    <strong>🎨 Правила хорошего тона:</strong>
    <ul>
        <li>Используйте не более <strong>3 уровней жирности</strong> на одной
странице</li>
        <li>Курсив — для <strong>коротких выделений</strong>, не для целых
абзацев</li>
        <li>Сочетайте с <strong>размером шрифта</strong> и <strong>цветом</strong>
для лучшего эффекта</li>
        <li>Проверяйте читаемость на <strong>разных устройствах</strong></li>
    </ul>
</div>
</div>
```

```
<!-- Лаборатория для экспериментов -->
<div class="лаборатория">
    <h2 style="color: white; margin-top: 0;">🔬 Лаборатория: Создайте свой стиль текста</h2>

    <div class="управление">
        <div>
            <h3 style="color: white;">Выберите жирность:</h3>
            <button class="кнопка-стиль" onclick="применитьСтиль('weight', '300')>300
(Light)</button>
            <button class="кнопка-стиль" onclick="применитьСтиль('weight', '400')>400
(Normal)</button>
            <button class="кнопка-стиль" onclick="применитьСтиль('weight', '600')>600
(Semi Bold)</button>
            <button class="кнопка-стиль" onclick="применитьСтиль('weight', '700')>700
(Bold)</button>
            <button class="кнопка-стиль" onclick="применитьСтиль('weight', '900')>900
(Black)</button>
        </div>
        <div>
            <h3 style="color: white;">Выберите наклон:</h3>
```

```

        <button class="кнопка-стиль" onclick="применитьСтиль('style',
'normal')">Normal</button>
        <button class="кнопка-стиль" onclick="применитьСтиль('style',
'italic')">Italic</button>
        <button class="кнопка-стиль" onclick="применитьСтиль('style',
'oblique')">Oblique</button>
    </div>

    <div>
        <h3 style="color: white;">Готовые стили:</h3>
        <button class="кнопка-стиль комбинированная"
onclick="применитьГотовыйСтиль('light-italic')">Light Italic</button>
        <button class="кнопка-стиль комбинированная"
onclick="применитьГотовыйСтиль('bold')">Заголовок</button>
        <button class="кнопка-стиль комбинированная"
onclick="применитьГотовыйСтиль('quote')">Цитата</button>
        <button class="кнопка-стиль комбинированная"
onclick="применитьГотовыйСтиль('warning')">Предупреждение</button>
    </div>
</div>

<div class="предпросмотр" id="предпросмотрТекста">
    Начните экспериментировать! Выберите стиль выше и увидите результат здесь.
</div>

<div style="margin-top: 25px; text-align: center;">
    <button class="кнопка-стиль" onclick="скопироватьКод()" style="background:
#f39c12;"> Скопировать CSS-код</button>
    <button class="кнопка-стиль" onclick="сброситьСтили()" style="background:
#7f8c8d;"> Сбросить</button>
</div>
</div>

<!-- Пример использования в реальном проекте --&gt;
&lt;div style="background: white; padding: 30px; border-radius: 15px; margin: 30px 0; border: 2px
solid #3498db;"&gt;
    &lt;h2 style="color: #2c3e50; margin-top: 0;"&gt;<img alt="Pencil icon" data-bbox="500 775 520 795"/> Пример: Оформление статьи</h2>

    <article>
        <h1 style="font-weight: 700; font-size: 36px; color: #2c3e50; margin-bottom: 20px;">
            Тайны старого сада
        </h1>

```

<p style="font-weight: 300; font-size: 18px; line-height: 1.7; color: #333; margin-bottom: 25px;">

Каждое утро, выходя в свой сад, я замечаю что-то новое. <**Сегодня распустилась первая роза**> —

алая, как утренняя заря. Её лепестки ещё хранят капли росы, которые переливаются на солнце всеми цветами радуги.

</p>

<p style="font-weight: 300; font-size: 18px; line-height: 1.7; color: #333; margin-bottom: 25px;">

<*«Сад — это не просто клумбы и грядки, — говорила мне бабушка, — это живая книга, где каждая страница — новое время года».*> Я тогда не понимала этих слов, но теперь, спустя годы, осознаю их мудрость.

</p>

<

## Советы начинающим садоводам

</h2>

<ul style="font-weight: 400; font-size: 17px; line-height: 1.6; color: #333; padding-left: 20px;">

<li style="margin-bottom: 10px;">

    <**Выбирайте растения**> по климату  
вашего региона

    </li>

    <li style="margin-bottom: 10px;">

        <**Не забывайте про полив**>,  
особенно в жаркие дни

        </li>

        <li style="margin-bottom: 10px;">

            <**Наблюдайте за своими**>  
растениями

— они сами подскажут, что им нужно

        </li>

</ul>

<div style="background: #f8f9fa; padding: 25px; border-radius: 10px; margin: 30px 0; border-left: 5px solid #2ecc71;">

<p style="font-weight: 500; font-style: italic; font-size: 20px; color: #27ae60; margin: 0;">

    Важно: Не переусердствуйте с удобрениями! Иногда растениям нужно просто дать расти в своём ритме.

</p>

```

        </div>

        <footer style="text-align: right; margin-top: 40px;">
            <p style="font-weight: 400; font-style: italic; color: #7f8c8d;">
                Статья написана <strong style="font-weight: 600;">Галиной
                Ивановной</strong>, садоводом с 40-летним стажем
            </p>
        </footer>
    </article>

    <div class="инфо-блок" style="margin-top: 30px;">
        <strong>❸ Что использовано в этом примере:</strong>
        <ul>
            <li>Заголовок h1: <code>font-weight: 700</code> (очень жирный)</li>
            <li>Основной текст: <code>font-weight: 300</code> (лёгкий, для удобного
                чтения)</li>
            <li>Важные слова: <code>font-weight: 600</code> (полужирный, для
                акцента)</li>
            <li>Цитата: <code>font-style: italic</code> + <code>font-weight: 400</code></li>
            <li>Важное замечание: Курсив + <code>font-weight: 500</code> + цветной
                фон</li>
            <li>Подпись: Курсив + жирное имя</li>
        </ul>
    </div>
</div>

<!-- Домашнее задание --&gt;
&lt;div style="background: linear-gradient(135deg, #667eea 0%, #764ba2 100%); color: white;
padding: 30px; border-radius: 15px; margin: 30px 0;"&gt;
    &lt;h2 style="color: white; margin-top: 0;"&gt;🏠 Домашнее задание&lt;/h2&gt;

    &lt;div style="display: grid; grid-template-columns: repeat(auto-fit, minmax(250px, 1fr)); gap:
20px; margin-top: 20px;"&gt;
        &lt;div style="background: rgba(255,255,255,0.1); padding: 20px; border-radius: 10px;
backdrop-filter: blur(10px);&gt;
            &lt;h3 style="color: white;"&gt;Задание 1: Оформите рецепт&lt;/h3&gt;
            &lt;p&gt;Создайте страницу с рецептом вашего любимого блюда. Используйте:&lt;/p&gt;
            &lt;ul&gt;
                &lt;li&gt;Название рецепта: &lt;strong&gt;900&lt;/strong&gt;&lt;/li&gt;
                &lt;li&gt;Ингредиенты: &lt;strong&gt;400&lt;/strong&gt;&lt;/li&gt;
                &lt;li&gt;Важные ингредиенты: &lt;strong&gt;600 + italic&lt;/strong&gt;&lt;/li&gt;
                &lt;li&gt;Советы: &lt;strong&gt;300 + italic&lt;/strong&gt;&lt;/li&gt;
            &lt;/ul&gt;
    &lt;/div&gt;
&lt;/div&gt;</pre>

```

```
<div style="background: rgba(255,255,255,0.1); padding: 20px; border-radius: 10px;  
backdrop-filter: blur(10px);">
```

```
    <h3 style="color: white;">Задание 2: Создайте объявление</h3>
```

```
    <p>Сделайте красивое объявление о пропаже кота/собаки:</p>
```

```
    <ul>
```

```
        <li>Заголовок "ПРОПАЛ": <strong>900</strong></li>
```

```
        <li>Описание: <strong>300</strong></li>
```

```
        <li>Контакты: <strong>700</strong></li>
```

```
        <li>Награда: <strong>800 + italic</strong></li>
```

```
    </ul>
```

```
</div>
```

```
<div style="background: rgba(255,255,255,0.1); padding: 20px; border-radius: 10px;  
backdrop-filter: blur(10px);">
```

```
    <h3 style="color: white;">Задание 3: Письмо внуку</h3>
```

```
    <p>Напишите письмо внуку о вашем детстве:</p>
```

```
    <ul>
```

```
        <li>Обращение: <strong>600 + italic</strong></li>
```

```
        <li>Воспоминания: <strong>300</strong></li>
```

```
        <li>Мораль: <strong>500 + italic</strong></li>
```

```
        <li>Подпись: <strong>400 + italic</strong></li>
```

```
    </ul>
```

```
</div>
```

```
</div>
```

```
<div style="text-align: center; margin-top: 25px;">
```

```
    <button class="кнопка-стиль" onclick="alert('Удачи в выполнении заданий! Помните:  
практика — лучший учитель.')">
```

```
        style="background: white; color: #764ba2; font-weight: 700;">
```

```
            🚀 Принять задание!
```

```
        </button>
```

```
</div>
```

```
</div>
```

```
<script>
```

```
    // Элемент для предпросмотра
```

```
    const предпросмотр = document.getElementById('предпросмотрТекста');
```

```
    let текущийВес = '400';
```

```
    let текущийСтиль = 'normal';
```

```
    // Функция применения стиля
```

```
    function применитьСтиль(тип, значение) {
```

```
        if (тип === 'weight') {
```

```
текущийВес = значение;
предпросмотр.style.fontWeight = значение;
} else if (тип === 'style') {
    текущийСтиль = значение;
    предпросмотр.style.fontStyle = значение;
}

предпросмотр.textContent = `Это текст с font-weight: ${текущийВес} и font-style: ${текущийСтиль}.

Попробуйте другие комбинации!`;

}

// Готовые стили
function применитьГотовыйСтиль(тип) {
    switch(тип) {
        case 'light-italic':
            предпросмотр.style.fontWeight = '300';
            предпросмотр.style.fontStyle = 'italic';
            предпросмотр.textContent = 'Нежное напоминание или поэтична строка.

Идеально для цитат.';
            break;

        case 'bold':
            предпросмотр.style.fontWeight = '700';
            предпросмотр.style.fontStyle = 'normal';
            предпросмотр.textContent = 'ВАЖНОЕ ОБЪЯВЛЕНИЕ! Заголовок, который
нельзя пропустить.';
            break;

        case 'quote':
            предпросмотр.style.fontWeight = '400';
            предпросмотр.style.fontStyle = 'italic';
            предпросмотр.textContent = '«Мудрость приходит с годами, но иногда
годы приходят одни». — Народная мудрость';
            break;

        case 'warning':
            предпросмотр.style.fontWeight = '800';
            предпросмотр.style.fontStyle = 'italic';
            предпросмотр.style.color = '#e74c3c';
            предпросмотр.textContent = 'ВНИМАНИЕ! Очень важное предупреждение,
которое нельзя игнорировать.';
            setTimeout(() => предпросмотр.style.color = '#333', 2000);
            break;
    }
}
```

```

        }

        текущийВес = предпросмотр.style.fontWeight;
        текущийСтиль = предпросмотр.style.fontStyle;
    }

    // Сброс стилей
    function сброситьСтили() {
        предпросмотр.style.fontWeight = '400';
        предпросмотр.style.fontStyle = 'normal';
        предпросмотр.style.color = '#333';
        предпросмотр.textContent = 'Начните экспериментировать! Выберите стиль выше и
увидите результат здесь。';
    }

    текущийВес = '400';
    текущийСтиль = 'normal';
}

// Копирование CSS-кода
function скопироватьКод() {
    const код = `font-weight: ${текущийВес};\nfont-style: ${текущийСтиль};`;
    navigator.clipboard.writeText(код)
        .then(() => alert('CSS-код скопирован в буфер обмена:\n' + код))
        .catch(err => alert('Не удалось скопировать код: ' + err));
}

// Инициализация
сброситьСтили();

</script>
</body>
</html>

```

## 4. Практические советы по использованию

### A. Для хорошей читаемости:

```

css

/* Основной текст */
.body-text {
    font-weight: 300; /* или 400 */
    font-size: 1.1rem;
}

```

```
line-height: 1.6;  
}  
  
/* Заголовки */  
h1, h2, h3 {  
    font-weight: 600; /* или 700 */  
}  
  
/* Важные моменты */  
.important {  
    font-weight: 600;  
    color: #d35400;  
}  
  
/* Цитаты */  
.quote {  
    font-style: italic;  
    font-weight: 400;  
    color: #7f8c8d;  
    padding-left: 20px;  
    border-left: 3px solid #3498db;  
}
```

## В. Чего избегать:

1. **Не используйте курсив для длинных текстов** — глаза быстро устают
2. **Не делайте весь текст жирным** — теряется акцентность
3. **Не смешивайте много разных жирностей** — максимум 3-4 уровня
4. **Проверяйте контраст** — тёмный текст на тёмном фоне не читается, даже если он жирный

## С. Сочетание с другими свойствами:

css

```
.красивый-акцент {  
    font-weight: 700;  
    font-style: italic;  
    color: #8b4513; /* Коричневый */  
    text-shadow: 1px 1px 2px rgba(0,0,0,0.1);  
    letter-spacing: 0.5px; /* Немного разрежаем буквы */  
}
```

## 5. Особенности работы с разными шрифтами

### Проверка поддержки:

css

```
/* Всегда указывайте fallback (запасной вариант) */
body {
    font-family: 'Roboto', Arial, sans-serif;
    font-weight: 300; /* Если Roboto Light не загрузится, будет Arial normal */
}
```

### Для Google Fonts:

При подключении шрифта указывайте нужные начертания:

html

```
<!-- Подключаем Roboto с разными весами и курсивом -->
<link href="https://fonts.googleapis.com/css2?
family=Roboto:ital,wght@0,300;0,400;0,700;1,300;1,400;1,700&display=swap" rel="stylesheet">
```

## 6. Быстрые команды для HTML (альтернатива CSS)

В HTML тоже есть теги для быстрого выделения, но лучше использовать CSS:

html

```
<!-- Устаревший способ (лучше не использовать) -->
<b>Жирный текст</b>
<i>Курсив</i>
```

```
<!-- Современный семантический способ -->
<strong>Важный текст (по умолчанию жирный)</strong>
<em>Акцентированный текст (по умолчанию курсив)</em>
```

```
<!-- Но лучше так: -->
<span class="жирный-текст">Текст с CSS-стилями</span>
```

### Почему CSS лучше:

- Больше контроля (можно использовать любую жирность, не только bold)

- Лучше для доступности (скринридеры правильно читают семантические теги)
- Легче изменять глобально

## 7. Итог: простое руководство

**Для начинающих:**

1. **Основной текст:** font-weight: 300 или 400
2. **Заголовки:** font-weight: 600 или 700
3. **Акценты:** font-weight: 500 или 600
4. **Цитаты:** font-style: italic + font-weight: 400
5. **Очень важное:** font-weight: 800 или 900 + возможно font-style: italic

**Простое правило:**

- **Жирность** = важность (чем жирнее, тем важнее)
- **Курсив** = особое отношение (цитата, мысль, термин)

**Запомните:** Сначала текст должен быть удобочитаемым, а уже потом красивым. Не жертвуйте читаемостью ради дизайна!

Теперь вы мастер выделения текста! Вы можете делать акценты, создавать иерархию важности и делать текст выразительным. В следующей главе мы научимся выравнивать текст — это ещё один важный инструмент для создания красивых страниц!

## ■ Выравнивание (text-align): Как в текстовом редакторе Word.

Дорогой друг, вы когда-нибудь обращали внимание, как текст располагается на странице? Одни абзацы выровнены по левому краю, другие — по центру, заголовки часто посередине, а в газетах текст иногда растягивается по всей ширине. Всё это управляет одним простым свойством CSS — text-align. Давайте научимся им пользоваться, как в привычном Word!

### 1. Что такое text-align?

text-align — это свойство CSS, которое определяет **горизонтальное выравнивание текста** внутри его контейнера.

Представьте, что у вас есть лист бумаги в клеточку (это ваш HTML-элемент), и вы пишете на нём текст. Свойство text-align говорит: «Расположи все строки этого текста относительно левого/правого края или по центру».

---

### 2. Основные значения text-align (как кнопки в Word!)

#### A. text-align: left; — По левому краю (самое частое)

css

```
p {  
    text-align: left;  
}
```

Как выглядит:

text

Вот пример текста, выровненного по левому краю. Все строки начинаются от одной вертикальной линии слева, а справа край неровный, «рваный».

## Когда использовать:

- Основной текст статей, блогов
- Сообщения в чатах (ваши сообщения)
- Любой текст для комфортного чтения
- **В языках с письмом слева направо (русский, английский)**

**Аналогия в Word:** Кнопка "Выровнять по левому краю" 

---

## B. text-align: right; — По правому краю

css

```
p {  
    text-align: right;  
}
```

### Как выглядит:

text

Вот пример текста,  
выровненного по правому краю.  
Все строки заканчиваются у одной вертикальной  
линии справа.

## Когда использовать:

- Даты в письмах и документах
- Цены в меню и прайс-листах
- Подписи в официальных документах
- **В языках с письмом справа налево (арабский, иврит)**

**Аналогия в Word:** Кнопка "Выровнять по правому краю" 

---

## C. text-align: center; — По центру

css

```
h1 {  
    text-align: center;
```

```
}
```

## Как выглядит:

text

Вот пример текста,  
выровненного по центру.

Каждая строка располагается  
посередине, на равном расстоянии  
от левого и правого края.

## Когда использовать:

- Заголовки и подзаголовки
- Логотипы и слоганы
- Цитаты и эпиграфы
- Приглашения и открытки
- Короткие важные сообщения

**Аналогия в Word:** Кнопка "Выровнять по центру" 

---

## D. text-align: justify; — По ширине (как в газетах)

css

```
p {  
    text-align: justify;  
}
```

## Как выглядит:

text

Вот пример текста, выровненного по ширине. Все строки, кроме последней, растянуты так, чтобы начинаться у левого края и заканчиваться у правого. Браузер добавляет пробелы между словами для выравнивания.

## Когда использовать:

- Газеты и журналы
- Книги и длинные статьи
- Официальные документы

- Когда нужен аккуратный, ровный правый край

**Важно:** Может создавать некрасивые большие промежутки между словами, особенно в узких колонках.

**Аналогия в Word:** Кнопка "Выровнять по ширине" 

---

## E. **text-align: start;** и **text-align: end;** — Умное выравнивание

css

```
p {  
    text-align: start; /* Как left для LTR, как right для RTL */  
    text-align: end; /* Как right для LTR, как left для RTL */  
}
```

**Что это:** Умные значения, которые автоматически учитывают направление текста (слева направо или справа налево).

**Когда использовать:**

- Для международных сайтов
  - Когда поддерживаются языки с разным направлением письма
- 

## 3. Наглядный пример всех вариантов

Создайте файл **выравнивание.html**:

```
html  
<!DOCTYPE html>  
<html lang="ru">  
<head>  
    <meta charset="UTF-8">  
    <meta name="viewport" content="width=device-width, initial-scale=1.0">  
    <title>Выравнивание текста как в Word</title>  
    <style>  
        /* Общие настройки */  
        * {
```

```
        box-sizing: border-box;
    }

body {
    font-family: 'Georgia', serif;
    line-height: 1.6;
    color: #333;
    background-color: #f8f9fa;
    padding: 30px;
    max-width: 1000px;
    margin: 0 auto;
}

h1 {
    text-align: center;
    color: #2c3e50;
    margin-bottom: 40px;
    padding-bottom: 15px;
    border-bottom: 3px solid #3498db;
}

/* Контейнер для примеров */
.раздел {
    background: white;
    padding: 30px;
    margin: 30px 0;
    border-radius: 15px;
    box-shadow: 0 5px 20px rgba(0,0,0,0.08);
}

.раздел h2 {
    color: #2c3e50;
    margin-top: 0;
    padding-bottom: 10px;
    border-bottom: 2px solid;
}

/* Примеры выравнивания */
.пример-выравнивания {
    margin: 25px 0;
    padding: 25px;
    border-radius: 10px;
    background: #f8f9fa;
    border: 2px solid #e9ecf;
```

```
position: relative;
}

.пример-выравнивания::before {
    content: "";
    position: absolute;
    top: 0;
    left: 0;
    right: 0;
    height: 5px;
    border-radius: 5px 5px 0 0;
}

/* Цвета для разных типов выравнивания */
.left::before { background: #e74c3c; }
.right::before { background: #3498db; }
.center::before { background: #2ecc71; }
.justify::before { background: #9b59b6; }

.пример-выравнивания h3 {
    margin-top: 0;
    color: #2c3e50;
    font-size: 20px;
}

.код-пример {
    font-family: 'Courier New', monospace;
    background: #2c3e50;
    color: white;
    padding: 15px;
    border-radius: 8px;
    margin: 15px 0;
    font-size: 16px;
}

/* Гид-линии для наглядности */
.гид-линии {
    position: relative;
    padding: 0 10px;
}

.гид-линии::after {
    content: "";
    position: absolute;
```

```
        top: 0;
        bottom: 0;
        left: 50%;
        width: 1px;
        background: rgba(52, 152, 219, 0.3);
        transform: translateX(-50%);
    }

.gid-linii::before {
    content: "";
    position: absolute;
    top: 0;
    bottom: 0;
    left: 0;
    width: 1px;
    background: rgba(231, 76, 60, 0.3);
}

/* Панель управления Word-стиль */
.panель-word {
    background: #2c3e50;
    color: white;
    padding: 30px;
    border-radius: 15px;
    margin: 30px 0;
}

.knopki-vyравнивания {
    display: flex;
    justify-content: center;
    gap: 10px;
    margin: 25px 0;
    flex-wrap: wrap;
}

.knopka-word {
    background: #34495e;
    color: white;
    border: none;
    width: 50px;
    height: 50px;
    border-radius: 8px;
    cursor: pointer;
    font-size: 20px;
}
```

```
display: flex;
align-items: center;
justify-content: center;
transition: all 0.3s;
}

.кнопка-word:hover {
    background: #3498db;
    transform: translateY(-3px);
}

.кнопка-word.active {
    background: #3498db;
    box-shadow: 0 0 0 2px white;
}

.поле-эксперимента {
    background: white;
    color: #333;
    padding: 30px;
    border-radius: 10px;
    margin-top: 25px;
    font-size: 18px;
    min-height: 150px;
    line-height: 1.6;
    border: 2px solid #3498db;
}

/*
Примеры из реальной жизни */
.реальный-пример {
    display: grid;
    grid-template-columns: repeat(auto-fit, minmax(300px, 1fr));
    gap: 25px;
    margin-top: 25px;
}

.карточка-примера {
    background: white;
    padding: 25px;
    border-radius: 10px;
    border: 2px solid #e9ecef;
}

/*
Информационные блоки */
```

```

.инфо-блок {
    background: #e8f4fc;
    padding: 20px;
    border-radius: 10px;
    margin: 20px 0;
    border-left: 5px solid #3498db;
}

.совет {
    background: #fff8e1;
    padding: 20px;
    border-radius: 10px;
    margin: 20px 0;
    border-left: 5px solid #f39c12;
}

/* Анимация для демонстрации */
@keyframes подсветка {
    0% { box-shadow: 0 0 0 rgba(52, 152, 219, 0.7); }
    70% { box-shadow: 0 0 10px rgba(52, 152, 219, 0); }
    100% { box-shadow: 0 0 0 rgba(52, 152, 219, 0); }
}

.подсветить {
    animation: подсветка 1.5s;
}

</style>
</head>
<body>

<h1> Выравнивание текста (text-align): Как в Microsoft Word</h1>

<div class="инфо-блок">
    <p><strong>Аналогия из Word:</strong> В текстовом редакторе Word есть 4 кнопки выравнивания —<br/> по левому краю, по центру, по правому краю и по ширине. В CSS эти же функции выполняет<br/> одно свойство <code>text-align</code> с соответствующими значениями.</p>
</div>

<!-- Раздел 1: Все варианты выравнивания -->
<div class="раздел">
    <h2>1. Все 4 варианта выравнивания</h2>

    <!-- По левому краю -->

```

```
<div class="пример-выравнивания left">
    <h3>text-align: left; — По левому краю</h3>
    <div class="код-пример">text-align: left;</div>
    <div class="гид-линии" style="text-align: left;">
        <p>Это самый распространённый способ выравнивания текста. Он естественен
```

для глаз,

поскольку мы читаем слева направо. Все строки начинаются от одной вертикальной линии,

создавая ровный левый край. Правый край при этом остаётся «рваным», но это не мешает чтению,

а наоборот — помогает глазу легко переходить с одной строки на другую.</p>

```
</div>
<div class="совет">
    <strong>17 Используйте для:</strong> основного текста статей, сообщений в блогах,
```

любых длинных текстов для чтения.

```
</div>
</div>
```

```
<!-- По правому краю -->
```

```
<div class="пример-выравнивания right">
    <h3>text-align: right; — По правому краю</h3>
    <div class="код-пример">text-align: right;</div>
    <div class="гид-линии" style="text-align: right;">
        <p>Этот вид выравнивания используется реже, но имеет свои специфические применения.
```

Он создаёт аккуратный правый край, в то время как левый остаётся неровным.

Читать такой текст труднее, поскольку глазам приходится каждый раз искать начало новой строки.

Однако для коротких текстовых блоков он может быть эффективным дизайнерским приёмом.</p>

```
</div>
<div class="совет">
    <strong>17 Используйте для:</strong> дат в письмах, подписей, цен в меню, выравнивания чисел в таблицах.
</div>
</div>
```

```
<!-- По центру -->
```

```
<div class="пример-выравнивания center">
    <h3>text-align: center; — По центру</h3>
    <div class="код-пример">text-align: center;</div>
    <div class="гид-линии" style="text-align: center;">
```

<p>Центрирование придаёт тексту официальный, торжественный вид. Каждая строка располагается

симметрично относительно центральной оси. Этот приём сразу привлекает внимание,

поэтому идеально подходит для заголовков, лозунгов и важных объявлений.

Однако для длинных текстов центрирование — плохой выбор: глаза быстро устают от постоянного

«перескакивания» с разной начальной точки.</p>

</div>

**<div class="совет">**

<strong>⌚ Используйте для:</strong> заголовков, подзаголовков, логотипов,

коротких цитат,

приглашений, открыток.

</div>

</div>

<!-- По ширине -->

**<div class="пример-выравнивания justify">**

<h3>text-align: justify; — По ширине (как в газетах)</h3>

**<div class="код-пример">text-align: justify;</div>**

**<div class="гид-линии" style="text-align: justify;">**

<p>Выравнивание по ширине создаёт аккуратные, ровные края с обеих сторон текстового блока.

Браузер автоматически регулирует пробелы между словами, растягивая строки на всю доступную ширину.

Это придаёт тексту официальный, строгий вид, характерный для газет, журналов и книг.

Однако в узких колонках могут появляться некрасивые большие пробелы между словами,

что ухудшает читаемость.</p>

</div>

**<div class="совет">**

<strong>📰 Используйте для:</strong> газетных колонок, книжных страниц, официальных документов,

когда нужен аккуратный вид с обоих краёв.

</div>

</div>

</div>

<!-- Раздел 2: Панель управления Word-стиль -->

**<div class="панель-word">**

<h2 style="color: white; margin-top: 0;">2. Виртуальный Word: Экспериментируйте!</h2>

<p style="color: #ecf0f1;">Нажмите кнопки выравнивания, как в настоящем Word, и смотрите, как меняется текст ниже.</p>

```
<div class="кнопки-выравнивания">
    <button class="кнопка-word active" id="btnLeft"
    onclick="изменитьВыравнивание('left')" title="По левому краю">
        
    </button>
    <button class="кнопка-word" id="btnCenter"
    onclick="изменитьВыравнивание('center')" title="По центру">
        
    </button>
    <button class="кнопка-word" id="btnRight" onclick="изменитьВыравнивание('right')"
    title="По правому краю">
        
    </button>
    <button class="кнопка-word" id="btnJustify" onclick="изменитьВыравнивание('justify')"
    title="По ширине">
        =
    </button>
</div>
```

<div class="поле-эксперимента" id="экспериментальноеПоле">

Дорогой друг! Это ваше поле для экспериментов с выравниванием текста.

Попробуйте все четыре варианта и посмотрите, как меняется восприятие текста.

Выравнивание по левому краю — самый естественный способ для чтения.

По центру — для заголовков и акцентов. По правому краю — для специальных случаев.

По ширине — для официальных документов и публикаций.

Вы можете даже изменить этот текст! Просто кликните по нему дважды и введите свой вариант.

</div>

```
<div style="margin-top: 20px; text-align: center;">
    <button class="кнопка-word" onclick="добавитьТекст()" style="width: auto; padding: 0
    20px;">
        ✨ Добавить ещё текст
    </button>
    <button class="кнопка-word" onclick="очиститьТекст()" style="width: auto; padding: 0
    20px;">
        🗑️ Очистить
    </button>
</div>
</div>
```

```
<!-- Раздел 3: Примеры из реальной жизни -->
<div class="раздел">
    <h2>3. Как в реальной жизни: Примеры использования</h2>

    <div class="реальный-пример">
        <!-- Пример 1: Письмо -->
        <div class="карточка-примера">
            <h3 style="text-align: center;">✉ Письмо другу</h3>
            <div style="text-align: right; color: #666; margin-bottom: 15px;">
                Москва, 15 марта 2024 г.
            </div>
            <div style="text-align: left; line-height: 1.7;">
                Дорогой Петр!<br><br>
                Пишу тебе это письмо, чтобы поделиться радостной новостью...
            </div>
            <div style="text-align: right; margin-top: 20px; font-style: italic;">
                С наилучшими пожеланиями,<br>
                <strong>Иван</strong>
            </div>
            <div class="код-пример" style="margin-top: 15px; font-size: 14px;">
                /* Дата справа, текст слева, подпись справа */
            </div>
        </div>

        <!-- Пример 2: Меню ресторана -->
        <div class="карточка-примера">
            <h3 style="text-align: center;">🍰 Меню ресторана</h3>
            <div style="margin: 20px 0;">
                <div style="display: flex; justify-content: space-between;">
                    <span style="text-align: left;">Борщ с пампушками</span>
                    <span style="text-align: right;">350 руб.</span>
                </div>
                <div style="display: flex; justify-content: space-between; margin-top: 10px;">
                    <span style="text-align: left;">Котлета по-киевски</span>
                    <span style="text-align: right;">420 руб.</span>
                </div>
                <div style="display: flex; justify-content: space-between; margin-top: 10px;">
                    <span style="text-align: left;">Компот из сухофруктов</span>
                    <span style="text-align: right;">150 руб.</span>
                </div>
            </div>
            <div class="код-пример" style="margin-top: 15px; font-size: 14px;">
                /* Названия слева, цены справа */
            </div>
        </div>
    </div>
```

```
</div>
</div>

<!-- Пример 3: Газетная статья --&gt;
&lt;div class="карточка-примера"&gt;
    &lt;h3 style="text-align: center;"&gt;📰 Газетная статья&lt;/h3&gt;
    &lt;div style="text-align: justify; line-height: 1.6; column-count: 2; column-gap: 20px;"&gt;
        Вчера в центральном парке города открылась новая детская площадка.
        Мэр лично присутствовал на церемонии открытия и пообщался с
        маленькими
        горожанами. Площадка оснащена современным безопасным
        оборудованием...
    &lt;/div&gt;
    &lt;div class="код-пример" style="margin-top: 15px; font-size: 14px;"&gt;
        /* Выравнивание по ширине в две колонки */
    &lt;/div&gt;
    &lt;/div&gt;
&lt;/div&gt;

<!-- Раздел 4: Практические задания --&gt;
&lt;div class="раздел"&gt;
    &lt;h2&gt;4. Практикум: Создайте свои документы&lt;/h2&gt;

    &lt;div class="инфо-блок"&gt;
        &lt;p&gt;&lt;strong&gt;Задание 1: Приглашение на юбилей&lt;/strong&gt;&lt;/p&gt;
        &lt;p&gt;Создайте красивое приглашение на 50-летие. Используйте:&lt;/p&gt;
        &lt;ul&gt;
            &lt;li&gt;Заголовок "Приглашение" — по центру, крупно&lt;/li&gt;
            &lt;li&gt;Текст приглашения — по ширине&lt;/li&gt;
            &lt;li&gt;Дата и время — по центру&lt;/li&gt;
            &lt;li&gt;Адрес — по левому краю&lt;/li&gt;
            &lt;li&gt;Подпись "С уважением, семья Ивановых" — по правому краю&lt;/li&gt;
        &lt;/ul&gt;
    &lt;/div&gt;

    &lt;div class="инфо-блок"&gt;
        &lt;p&gt;&lt;strong&gt;Задание 2: Распорядок дня&lt;/strong&gt;&lt;/p&gt;
        &lt;p&gt;Создайте распорядок дня пенсионера:&lt;/p&gt;
        &lt;ul&gt;
            &lt;li&gt;Время (08:00, 09:00 и т.д.) — по правому краю&lt;/li&gt;
            &lt;li&gt;Деятельность (зарядка, завтрак) — по левому краю&lt;/li&gt;
            &lt;li&gt;Заголовок "Мой идеальный день" — по центру&lt;/li&gt;
        &lt;/ul&gt;
    &lt;/div&gt;
</pre>
```

```
</div>

<div class="инфо-блок">
    <p><strong>Задание 3: Объявление о пропаже</strong></p>
    <p>Создайте объявление о пропаже кота:</p>
    <ul>
        <li>"ПРОПАЛ КОТ" — по центру, очень крупно</li>
        <li>Описание кота — по левому краю</li>
        <li>Особые приметы — по ширине</li>
        <li>Контакты — по правому краю</li>
        <li>"НАГРАДА" — по центру, жирно</li>
    </ul>
</div>
</div>

<!-- Раздел 5: Важные нюансы --&gt;
&lt;div class="раздел"&gt;
    &lt;h2&gt;5. Важные нюансы и советы&lt;/h2&gt;

    &lt;div class="совет"&gt;
        &lt;h3&gt;⚠ text-align работает только на блоковые элементы!&lt;/h3&gt;
        &lt;p&gt;Если применить &lt;code&gt;text-align&lt;/code&gt; к строчному элементу (например,
&lt;code&gt;&amp;lt;span&amp;gt;&lt;/code&gt;),
            он не сработает. Нужно сделать его блочным:&lt;/p&gt;
        &lt;div class="код-пример"&gt;
/* Так НЕ сработает: */
span {
    text-align: center; /* Не работает! */
}

/* А так сработает: */
span {
    display: block; /* или inline-block */
    text-align: center; /* Теперь работает! */
}

&lt;/div&gt;
&lt;/div&gt;

&lt;div class="совет"&gt;
    &lt;h3&gt;⌚ text-align vs margin: auto для центрирования блоков&lt;/h3&gt;
    &lt;p&gt;Не путайте!&lt;/p&gt;
    &lt;div class="код-пример"&gt;
/* Выравнивание ТЕКСТА внутри блока: */
div {</pre>
```

```
text-align: center; /* Текст по центру блока */  
}  
  
/* Выравнивание самого БЛОКА на странице: */  
div {  
    width: 500px;  
    margin: 0 auto; /* Блок по центру страницы */  
    text-align: left; /* А текст внутри — слева */  
}  
    </div>  
    </div>  
  
<div class="совет">  
    <h3>⚠️ Проблемы с justify (выравниванием по ширине)</h3>  
    <p>При выравнивании по ширине могут возникать:</p>  
    <ul>  
        <li>«Реки» — пустые вертикальные промежутки</li>  
        <li>Слишком большие пробелы между словами</li>  
        <li>Нечитаемые последние строки</li>  
    </ul>  
    <p><strong>Решение:</strong> Используйте <code>hyphens: auto;</code> для  
переноса слов:</p>  
    <div class="код-пример">  
p {  
    text-align: justify;  
    hyphens: auto; /* Автоматические переносы слов */  
    text-justify: inter-word; /* Лучшее распределение пробелов */  
}  
    </div>  
    </div>  
    </div>  
  
<!-- Шпаргалка -->  
<div style="background: linear-gradient(135deg, #667eea 0%, #764ba2 100%); color: white;  
padding: 30px; border-radius: 15px; margin: 30px 0;">  
    <h2 style="color: white; text-align: center;">📋 Шпаргалка: Когда что использовать</h2>  
  
<div style="display: grid; grid-template-columns: repeat(auto-fit, minmax(200px, 1fr)); gap:  
20px; margin-top: 25px;">  
    <div style="background: rgba(255,255,255,0.1); padding: 20px; border-radius: 10px;  
backdrop-filter: blur(10px);">  
        <div style="text-align: center; font-size: 30px; margin-bottom: 10px;">✍</div>  
        <h3 style="color: white; text-align: center;">text-align: left</h3>  
        <ul style="font-size: 14px;">
```

```
<li>Основной текст</li>
<li>Сообщения в чате</li>
<li>Любые длинные тексты</li>
<li>По умолчанию для LTR</li>
</ul>
</div>

<div style="background: rgba(255,255,255,0.1); padding: 20px; border-radius: 10px; backdrop-filter: blur(10px);">
    <div style="text-align: center; font-size: 30px; margin-bottom: 10px;">✉</div>
    <h3 style="color: white; text-align: center;">text-align: center</h3>
    <ul style="font-size: 14px;">
        <li>Заголовки</li>
        <li>Логотипы</li>
        <li>Цитаты</li>
        <li>Приглашения</li>
    </ul>
</div>

<div style="background: rgba(255,255,255,0.1); padding: 20px; border-radius: 10px; backdrop-filter: blur(10px);">
    <div style="text-align: center; font-size: 30px; margin-bottom: 10px;">✉</div>
    <h3 style="color: white; text-align: center;">text-align: right</h3>
    <ul style="font-size: 14px;">
        <li>Даты</li>
        <li>Цены</li>
        <li>Подписи</li>
        <li>Числа в таблицах</li>
    </ul>
</div>

<div style="background: rgba(255,255,255,0.1); padding: 20px; border-radius: 10px; backdrop-filter: blur(10px);">
    <div style="text-align: center; font-size: 30px; margin-bottom: 10px;">≡</div>
    <h3 style="color: white; text-align: center;">text-align: justify</h3>
    <ul style="font-size: 14px;">
        <li>Газеты</li>
        <li>Книги</li>
        <li>Официальные документы</li>
        <li>Колонки текста</li>
    </ul>
</div>
</div>
```

```
<div style="text-align: center; margin-top: 30px;">
    <div style="background: rgba(255,255,255,0.2); padding: 15px; border-radius: 10px;
display: inline-block;">
        <strong>Золотое правило:</strong> <code>text-align: left;</code> для чтения,
        <code>text-align: center;</code> для акцента, остальное — для специальных
случаев!
    </div>
</div>
</div>

<script>
// Элементы для управления
const экспериментальноеПоле = document.getElementById('экспериментальноеПоле');
const кнопки = {
    left: document.getElementById('btnLeft'),
    center: document.getElementById('btnCenter'),
    right: document.getElementById('btnRight'),
    justify: document.getElementById('btnJustify')
};

// Функция изменения выравнивания
function изменитьВыравнивание(тип) {
    // Применяем выравнивание
    экспериментальноеПоле.style.textAlign = тип;

    // Подсвечиваем активную кнопку
    Object.values(кнопки).forEach(btn => btn.classList.remove('active'));
    кнопки[тип].classList.add('active');

    // Анимация подсветки
    экспериментальноеПоле.classList.remove('подсветить');
    void экспериментальноеПоле.offsetWidth; // Перезапуск анимации
    экспериментальноеПоле.classList.add('подсветить');

    // Показываем подсказку
    const подсказки = {
        left: 'Текст выровнен по левому краю — идеально для чтения!',
        center: 'Текст по центру — отлично для заголовков и акцентов!',
        right: 'Текст по правому краю — подходит для дат и подписей!',
        justify: 'Текст по ширине — как в газетах и книгах!'
    };

    экспериментальноеПоле.title = подсказки[тип];
}

}
```

```

// Функция добавления текста
function добавитьТекст() {
    const дополнительныеТексты = [
        "\n\nВыравнивание текста — одна из самых важных основ веб-тиографики. " +
        "Правильно выбранное выравнивание делает текст не только красивым, но и удобочитаемым.",
        "\n\nПомните: читабельность всегда важнее дизайна. " +
        "Даже самый красивый текст бесполезен, если его невозможно прочитать.",
        "\n\nЭкспериментируйте с разными видами выравнивания для разных типов контента. " +
        "Каждая ситуация требует своего подхода!"
    ];
}

const случайныйТекст = дополнительныеТексты[Math.floor(Math.random() * дополнительныеТексты.length)];
экспериментальноеПоле.textContent += случайныйТекст;
}

// Функция очистки текста
function очиститьТекст() {
    if (confirm('Очистить поле для экспериментов?')) {
        экспериментальноеПоле.textContent = 'Кликните дважды, чтобы редактировать этот текст.';
        изменитьВыравнивание('left');
    }
}

// Редактирование текста по двойному клику
экспериментальноеПоле.addEventListener('dblclick', function() {
    const старыйТекст = this.textContent;
    const новыйТекст = prompt('Введите свой текст для экспериментов:', старыйТекст);

    if (новыйТекст !== null && новыйТекст.trim() !== '') {
        this.textContent = новыйТекст;
    }
});

// Инициализация
изменитьВыравнивание('left');

</script>

```

```
</body>  
</html>
```

## 6. Практические примеры для пенсионеров

### Пример 1: Поздравительная открытка

html

```
<!DOCTYPE html>  
<html>  
<head>  
    <style>  
        .открытка {  
            width: 500px;  
            margin: 0 auto;  
            padding: 40px;  
            background: #fff9f0;  
            border: 3px solid #d4a762;  
            border-radius: 15px;  
            font-family: 'Georgia', serif;  
        }  
  
        .заголовок {  
            text-align: center;  
            font-size: 32px;  
            color: #8b4513;  
            margin-bottom: 30px;  
        }  
  
        .текст {  
            text-align: justify;  
            font-size: 20px;  
            line-height: 1.7;  
            color: #333;  
            margin-bottom: 25px;  
        }  
  
        .подпись {  
            text-align: right;  
            font-style: italic;  
            color: #666;  
        }  
    </style>  
</head>  
<body>  
    <div class="открытка">  
        <h1 class="заголовок">С Днем  
Победы!</h1>  
        <p class="текст">Поздравляем вас с  
великой победой в Великой Отечественной  
войне! Желаем вам здоровья, счастья и  
 успехов в жизни.  
        <p class="подпись">С любовью,  
Ваша семья</p>  
    </div>  
</body>  
</html>
```

```

        font-size: 18px;
    }

    .дата {
        text-align: right;
        color: #888;
        margin-top: 10px;
    }

```

</style>

</head>

<body>

<div class="открытика">

<h1 class="заголовок">С Днём рождения!</h1>

<p class="текст">

Дорогая Мария Ивановна! От всей души поздравляем Вас с юбилеем!  
 Желаем крепкого здоровья, бодрости духа, радости в каждом дне и  
 внимания близких. Пусть Ваша мудрость и доброта ещё много лет  
 согревают всех вокруг!

</p>

<p class="подпись">

С любовью и уважением,<br>

Ваши соседи по даче

</p>

<div class="дата">

15 марта 2024 года

</div>

</div>

</body>

</html>

## Пример 2: Расписание приёма лекарств

html

```

<!DOCTYPE html>
<html>
<head>
<style>
    .расписание {
        width: 400px;
        margin: 0 auto;

```

```
        font-family: Arial, sans-serif;
    }

h1 {
    text-align: center;
    color: #2c3e50;
    margin-bottom: 30px;
}

.прием {
    display: flex;
    justify-content: space-between;
    margin-bottom: 15px;
    padding: 15px;
    background: #f8f9fa;
    border-radius: 8px;
    border-left: 5px solid #3498db;
}

.время {
    text-align: right;
    font-weight: bold;
    color: #2c3e50;
    min-width: 80px;
}

.лекарство {
    text-align: left;
    flex-grow: 1;
    padding-left: 20px;
}

.важно {
    text-align: center;
    background: #fff3cd;
    border: 1px solid #ffea7;
    padding: 10px;
    margin-top: 20px;
    border-radius: 5px;
    font-weight: bold;
}

</style>
</head>
<body>
```

```
<div class="расписание">
    <h1> Расписание приёма лекарств</h1>

    <div class="прием">
        <div class="время">08:00</div>
        <div class="лекарство">Таблетка от давления</div>
    </div>

    <div class="прием">
        <div class="время">12:00</div>
        <div class="лекарство">Витамины</div>
    </div>

    <div class="прием">
        <div class="время">18:00</div>
        <div class="лекарство">Таблетка от давления</div>
    </div>

    <div class="прием">
        <div class="время">22:00</div>
        <div class="лекарство">Успокоительное</div>
    </div>

    <div class="важно">
         Не забывайте запивать лекарства водой!
    </div>
</div>
</body>
</html>
```

## 7. Итог: Простые правила для пенсионеров

### Запомните 4 основных значения:

1. left — по левому краю (для обычного текста)
2. center — по центру (для заголовков)
3. right — по правому краю (для дат и подписей)
4. justify — по ширине (для официальных документов)

## Простые аналогии:

- **Как в письме другу:** Текст слева, дата справа, подпись справа
- **Как в меню:** Блюда слева, цены справа
- **Как в газете:** Весь текст по ширине
- **Как в приглашении:** Всё по центру

## Золотые правила:

1. **Основной текст всегда** `text-align: left` — так легче всего читать
2. **Заголовки** — `text-align: center` — привлекают внимание
3. **Даты и цифры** — `text-align: right` — выглядят аккуратно
4. **Используйте** `justify` **осторожно** — может создавать некрасивые пробелы

## Простой способ запомнить:

css

```
/* Для почти всего */  
p { text-align: left; }  
  
/* Для заголовков */  
h1, h2, h3 { text-align: center; }  
  
/* Для дат и подписей */  
.дата, .подпись { text-align: right; }  
  
/* Для официальных бумаг */  
.документ { text-align: justify; }
```

**Теперь вы знаете всё о выравнивании текста!** Это один из самых простых, но самых важных инструментов CSS. Попробуйте создать свою открытку, письмо или расписание — и вы увидите, как изменится восприятие текста в зависимости от выравнивания!

В следующей главе мы перейдём к очень важной теме — работе с блоками и отступами. Это основа всей вёрстки страниц!

## ■ Практикум №2: Пишем красивое поздравительное письмо в веб-формате.

Дорогой друг, давайте создадим настоящее произведение искусства — цифровое поздравительное письмо, которое можно отправить ссылкой, распечатать или показать на экране компьютера. Мы применим ВСЁ, что уже изучили: цвета, шрифты, размеры, жирность, курсив и выравнивание!

### ШАГ 1: Подготовка «рабочего места»

#### 1.1. Создайте папку и файлы

1. На Рабочем столе создайте папку **ПоздравительноеПисьмо**
2. Внутри создайте два файла:

- `index.html` — структура письма
  - `style.css` — оформление письма
- 

### ШАГ 2: Создаём HTML-строктуру письма

#### Файл `письмо.html`:

```
html
<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Поздравительное письмо от Галины Ивановны</title>
    <!-- Подключаем красивые шрифты из Google -->
    <link rel="preconnect" href="https://fonts.googleapis.com">
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
    <link href="https://fonts.googleapis.com/css2?
family=Cormorant+Garamond:wght@400;600;700&family=Marck+Script&family=Open+Sans:wght@300;4
00;600&display=swap" rel="stylesheet">
    <!-- Подключаем наши стили -->
    <link rel="stylesheet" href="style.css">
```

```
<!-- Иконка конверта для красоты -->
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/
all.min.css">
</head>
<body>
    <div class="конверт">
        <div class="конверт-уголок"></div>
    </div>

    <div class="письмо">
        <!-- Шапка письма с датой и местом -->
        <div class="шапка">
            <div class="место">
                <i class="fas fa-map-marker-alt"></i> г. Москва
            </div>
            <div class="дата">
                <i class="far fa-calendar"></i> 15 марта 2024 года
            </div>
        </div>

        <!-- Приветствие (самая верхняя часть) -->
        <div class="приветствие">
            <h1>Дорогая внучка Анечка!</h1>
            <div class="декоративная-линия"></div>
        </div>

        <!-- Основной текст письма -->
        <div class="основной-текст">
            <p class="абзац первый-абзац">
                Пишу тебе это письмо в прекрасный весенний день, когда за окном поют
                птицы,
                а солнышко ласково греет землю после долгой зимы. Сижу на кухне за
                большим
                дубовым столом, который помнит ещё твоего прадеда, пью ароматный чай с
                мяты
                и вспоминаю, какой ты была маленькой.
            </p>

            <p class="абзац">
                Помнишь, как ты в пять лет помогала мне печь пироги? Весь кухонный стол
                был
                в муке, а у тебя на носу белая точка, как у снегиря. <span class="выделенный-
                текст">Ты так старалась,

```

так хотела сделать всё правильно</span>, что у меня до сих пор щемит сердце от этой нежности. Те пироги, может, и не были идеальными по форме, но были самыми вкусными на свете — потому что сделаны с любовью.

</p>

```
<div class="цитата">
    <div class="кавычки"><</div>
    <p>Жизнь — как тот пирог: иногда немного кривой, иногда подгорает с краю, но всегда прекрасна, если делишь её с близкими.</p>
    <div class="кавычки"><</div>
</div>
```

<p class="абзац">

Сейчас ты уже взрослая, самостоятельная женщина. Я вижу твои успехи, горжусь тобой и иногда даже завидую твоей энергии! <span class="важное">Но помни, пожалуйста,

самое главное: не забывай отдыхать, береги своё здоровье и находи время для простых радостей</span> — чашки чая на закате, прогулки в парке, разговоров

по душам.

</p>

<p class="абзац">

На даче уже проклонулись первые цветы — тюльпаны, которые мы с тобой сажали прошлой осенью. <i class="fas fa-seedling" style="color: #27ae60;"></i>

Они такие же яркие и жизнерадостные, как ты.

Когда приедешь летом в гости, обязательно покажу тебе наш новый розарий

—

посадила сорта с твоими любимыми названиями: «Анютка», «Улыбка», «Нежность».

</p>

</div>

<!-- Пожелания в отдельном блоке -->

<div class="пожелания">

<h2><i class="far fa-heart"></i> Мои пожелания тебе:</h2>

<ul>

- <i class="fas fa-star"></i> Крепкого здоровья — это главное богатство!
- <i class="fas fa-star"></i> Любви — чтобы сердце всегда было теплым!
- <i class="fas fa-star"></i> Терпения — в жизни бывает всякое!
- <i class="fas fa-star"></i> Радости — в больших и маленьких моментах!
- <i class="fas fa-star"></i> Успехов в работе — но без переутомления!

</ul>

```
</div>

<!-- Прощание и подпись -->
<div class="прощание">
    <p>
        Пиши, как у тебя дела. Звони, когда есть минутка. <span class="теплые-слова">Знай, что ты всегда
        в моих мыслях и молитвах.</span>
    </p>

    <div class="подпись">
        <div class="подпись-текст">
            Целую крепко, обнимаю<br>
            <span class="имя">Твоя бабушка Галина</span>
        </div>
        <div class="подпись-факсимиле">
            <i class="fas fa-pen-fancy"></i>
        </div>
    </div>

    <div class="постскриптум">
        <p><strong>P.S.</strong> Посылаю тебе рецепт тех самых пирогов. Испекла
        вчера —
        как в старые времена. Хранится в конверте на обратной стороне этого
        письма.</p>
        </div>
    </div>

    <!-- Декоративные элементы -->
    <div class="декорации">
        <div class="цветок цветок-1"><i class="fas fa-leaf"></i></div>
        <div class="цветок цветок-2"><i class="fab fa-pagelines"></i></div>
        <div class="цветок цветок-3"><i class="fas fa-spa"></i></div>
    </div>
</div>

<!-- Рецепт пирогов (появляется при клике) -->
<div class="рецепт-конверт" id="рецептКонверт">
    <div class="рецепт" id="рецепт">
        <button class="закрыть-рецепт" onclick="скрытьРецепт()">
            <i class="fas fa-times"></i>
        </button>
        <h2><i class="fas fa-utensils"></i> Семейный рецепт пирогов с яблоками</h2>
        <div class="рецепт-содержимое">
```

```

<div class="ингредиенты">
    <h3><i class="fas fa-shopping-basket"></i> Ингредиенты:</h3>
    <ul>
        <li>Мука — 500 г (просеять!)</li>
        <li>Сливочное масло — 200 г</li>
        <li>Сметана — 150 г</li>
        <li>Яйца — 2 шт.</li>
        <li>Сахар — 150 г</li>
        <li>Яблоки — 5-6 шт. (антоновка)</li>
        <li>Корица — по вкусу</li>
        <li>Ванильный сахар — 1 пакетик</li>
        <li>Щепотка соли</li>
    </ul>
</div>
<div class="приготовление">
    <h3><i class="fas fa-mortar-pestle"></i> Приготовление:</h3>
    <ol>
        <li>Масло растереть с сахаром до крема</li>
        <li>Добавить яйца, сметану, соль, перемешать</li>
        <li>Постепенно ввести муку, замесить тесто</li>
        <li>Убрать в холодильник на 30 минут</li>
        <li>Яблоки очистить, нарезать дольками</li>
        <li>Тесто раскатать, выложить в форму</li>
        <li>Сверху разложить яблоки, посыпать корицей</li>
        <li>Выпекать 40 минут при 180°C</li>
    </ol>
    <p class="совет-рецепта">
        <i class="fas fa-lightbulb"></i> <strong>Секрет бабушки:</strong>
    
```

Добавь в тесто

щепотку лимонной цедры — пирог получится ароматнее!

```

</p>
</div>
</div>
</div>

```

<!-- Кнопка для показа рецепта -->

```

<button class="кнопка-рецепта" onclick="показатьРецепт()">
    <i class="fas fa-envelope-open-text"></i> Открыть рецепт пирогов
</button>

```

<!-- Подсказка внизу -->

```

<div class="подсказка">
    <p><i class="fas fa-mouse-pointer"></i> Это интерактивное письмо! Можно:</p>

```

```

<ul>
    <li>Изменить цвета в CSS-файле</li>
    <li>Добавить свою фотографию</li>
    <li>Распечатать (Ctrl+P)</li>
    <li>Отправить ссылку родным</li>
</ul>
</div>

<script>
function показатьРецепт() {
    document.getElementById('рецептКонверт').style.display = 'flex';
    document.getElementById('рецепт').classList.add('показать');
}

function скрытьРецепт() {
    document.getElementById('рецептКонверт').style.display = 'none';
    document.getElementById('рецепт').classList.remove('показать');
}

// Закрытие рецепта при клике вне его
document.getElementById('рецептКонверт').addEventListener('click', function(e) {
    if (e.target === this) {
        скрытьРецепт();
    }
});
</script>
</body>
</html>

```

---

## ШАГ 3: Создаём волшебные стили для письма

**Файл style.css:**

```

css
=====
ВОЛШЕБНОЕ ПОЗДРАВИТЕЛЬНОЕ ПИСЬМО
Все значения можно менять как угодно!
===== */

/* 1. ОСНОВНЫЕ НАСТРОЙКИ СТРАНИЦЫ */
* {

```

```
    box-sizing: border-box;
    margin: 0;
    padding: 0;
}

body {
    font-family: 'Open Sans', sans-serif;
    line-height: 1.6;
    color: #333;
    background: linear-gradient(135deg, #f5f7fa 0%, #e4efe9 100%);
    min-height: 100vh;
    display: flex;
    flex-direction: column;
    align-items: center;
    justify-content: center;
    padding: 30px;
    position: relative;
}
```

/\* 2. ДЕКОРАТИВНЫЙ КОНВЕРТ (фон) \*/

```
.конверт {
    position: fixed;
    top: 50%;
    left: 50%;
    transform: translate(-50%, -50%);
    width: 90vw;
    max-width: 900px;
    height: 90vh;
    max-height: 1200px;
    background: #ffffaf0;
    border: 15px solid #f8e8d0;
    border-radius: 5px;
    z-index: -1;
    box-shadow:
        0 20px 60px rgba(139, 69, 19, 0.1),
        inset 0 0 100px rgba(255, 248, 225, 0.5);
}
```

```
.конверт-уголок {
    position: absolute;
    top: -15px;
    right: -15px;
    width: 100px;
    height: 100px;
```

```
background: #d4a762;
clip-path: polygon(100% 0, 0 0, 100% 100%);
opacity: 0.7;
}
```

```
/* 3. ОСНОВНОЕ ПИСЬМО */
```

```
.письмо {
    width: 85vw;
    max-width: 800px;
    background: white;
    padding: 50px;
    border-radius: 3px;
    box-shadow:
        0 10px 40px rgba(0, 0, 0, 0.08),
        0 0 1px rgba(139, 69, 19, 0.05);
    position: relative;
    z-index: 1;
    margin: 30px auto;
    border-left: 8px solid #d4a762;
}
```

```
/* 4. ШАПКА ПИСЬМА (дата и место) */
```

```
.шапка {
    display: flex;
    justify-content: space-between;
    margin-bottom: 40px;
    padding-bottom: 20px;
    border-bottom: 2px dashed #e9d4b3;
    font-family: 'Cormorant Garamond', serif;
    font-size: 18px;
    color: #8b4513;
}
```

```
.место, .дата {
    display: flex;
    align-items: center;
    gap: 10px;
}
```

```
.место i, .дата i {
    color: #d4a762;
    font-size: 20px;
}
```

```
/* 5. ПРИВЕТСТВИЕ */

.приветствие {
    text-align: center;
    margin-bottom: 40px;
}

.приветствие h1 {
    font-family: 'Marck Script', cursive;
    font-size: 48px;
    color: #8b4513;
    margin-bottom: 20px;
    text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.05);
    font-weight: 400;
}

.декоративная-линия {
    width: 200px;
    height: 3px;
    background: linear-gradient(90deg, transparent, #d4a762, transparent);
    margin: 0 auto;
}

/* 6. ОСНОВНОЙ ТЕКСТ */

.основной-текст {
    font-family: 'Cormorant Garamond', serif;
    font-size: 22px;
    line-height: 1.8;
    color: #5d4037;
    text-align: justify;
    hyphens: auto;
}

.абзац {
    margin-bottom: 30px;
    text-indent: 40px;
}

.первый-абзац::first-letter {
    font-family: 'Marck Script', cursive;
    font-size: 42px;
    color: #8b4513;
    float: left;
    line-height: 1;
    margin-right: 8px;
}
```

```
margin-top: 5px;
}

.выделенный-текст {
    font-weight: 600;
    color: #8b4513;
    font-style: italic;
    background: linear-gradient(120deg, #fff9c4 0%, #fff9c4 100%);
    background-repeat: no-repeat;
    background-size: 100% 0.4em;
    background-position: 0 88%;
}

.важное {
    font-weight: 700;
    color: #d35400;
    border-left: 4px solid #e67e22;
    padding-left: 15px;
    display: inline-block;
}

/* 7. ЦИТАТА */

.цитата {
    background: #f9f5f0;
    padding: 30px;
    margin: 40px 0;
    border-radius: 10px;
    border-left: 5px solid #d4a762;
    position: relative;
    font-style: italic;
    font-size: 20px;
    color: #7d5d3b;
}

.кавычки {
    font-family: 'Times New Roman', serif;
    font-size: 80px;
    color: #d4a762;
    position: absolute;
    line-height: 1;
}

.кавычки:first-child {
    top: -10px;
}
```

```
    left: 10px;  
}  
  
.кавычки:last-child {  
    bottom: -40px;  
    right: 10px;  
}  
  
.цитата p {  
    margin: 0;  
    padding: 0 30px;  
    text-align: center;  
}  
  
/* 8. ПОЖЕЛАНИЯ */  
.пожелания {  
    background: linear-gradient(135deg, #fff9e6 0%, #fff0cc 100%);  
    padding: 30px;  
    border-radius: 15px;  
    margin: 40px 0;  
    border: 2px solid #ffd166;  
    box-shadow: 0 5px 15px rgba(255, 209, 102, 0.1);  
}  
  
.пожелания h2 {  
    font-family: 'Cormorant Garamond', serif;  
    font-size: 28px;  
    color: #8b4513;  
    margin-bottom: 25px;  
    text-align: center;  
    display: flex;  
    align-items: center;  
    justify-content: center;  
    gap: 15px;  
}  
  
.пожелания h2 i {  
    color: #e74c3c;  
}  
  
.пожелания ul {  
    list-style: none;  
    padding-left: 20px;  
}
```

```
.пожелания li {
    font-size: 20px;
    margin-bottom: 15px;
    padding-left: 10px;
    display: flex;
    align-items: center;
    gap: 15px;
    color: #5d4037;
}

.пожелания li i {
    color: #f39c12;
    font-size: 16px;
}

/* 9. ПРОЩАНИЕ И ПОДПИСЬ */

.прощание {
    margin-top: 50px;
    padding-top: 30px;
    border-top: 2px dashed #e9d4b3;
}

.теплые-слова {
    font-weight: 600;
    color: #8b4513;
    background: url('data:image/svg+xml;utf8,<svg xmlns="http://www.w3.org/2000/svg" width="20" height="10" viewBox="0 0 20 10"><path d="M0,5 Q5,0 10,5 T20,5" stroke="%23d4a762" fill="none" stroke-width="0.5"/></svg>') repeat-x bottom;
    padding-bottom: 3px;
}

.подпись {
    display: flex;
    justify-content: flex-end;
    align-items: center;
    gap: 30px;
    margin: 40px 0;
}

.подпись-текст {
    text-align: right;
    font-family: 'Marck Script', cursive;
    font-size: 28px;
}
```

```
color: #8b4513;
line-height: 1.4;
}

.имя {
    font-size: 32px;
    font-weight: bold;
    display: block;
    margin-top: 10px;
}

.подпись-факсимиле {
    font-size: 60px;
    color: #d4a762;
    opacity: 0.7;
}

.постскриптум {
    background: #f8f9fa;
    padding: 25px;
    border-radius: 10px;
    margin-top: 40px;
    border-left: 5px solid #3498db;
    font-size: 18px;
    color: #2c3e50;
}

.постскриптум strong {
    color: #2980b9;
}

/* 10. ДЕКОРАТИВНЫЕ ЭЛЕМЕНТЫ */

.декорации {
    position: absolute;
    top: 0;
    right: 0;
    bottom: 0;
    left: 0;
    pointer-events: none;
    z-index: -1;
}

.цветок {
    position: absolute;
```

```
    color: rgba(212, 167, 98, 0.2);
    font-size: 40px;
}
```

```
.цветок-1 {
    top: 10%;
    left: 5%;
    transform: rotate(45deg);
}
```

```
.цветок-2 {
    bottom: 15%;
    right: 8%;
    transform: rotate(-20deg);
}
```

```
.цветок-3 {
    top: 40%;
    right: 5%;
    transform: rotate(15deg);
}
```

```
/* 11. РЕЦЕПТ ПИРОГОВ (скрытый до клика) */
```

```
.рецепт-конверт {
    position: fixed;
    top: 0;
    left: 0;
    width: 100%;
    height: 100%;
    background: rgba(0, 0, 0, 0.7);
    display: none;
    align-items: center;
    justify-content: center;
    z-index: 1000;
    padding: 20px;
}
```

```
.рецепт {
    background: white;
    width: 90%;
    max-width: 800px;
    max-height: 90vh;
    overflow-y: auto;
    padding: 40px;
```

```
border-radius: 20px;  
box-shadow: 0 20px 60px rgba(0, 0, 0, 0.3);  
border: 1px solid #fff9f0;  
position: relative;  
transform: translateY(50px);  
opacity: 0;  
transition: all 0.5s ease;  
}  
  
}

.рецепт.показать {
```

```
    transform: translateY(0);  
    opacity: 1;  
}
```

```
.закрыть-рецепт {  
    position: absolute;  
    top: 20px;  
    right: 20px;  
    background: #e74c3c;  
    color: white;  
    border: none;  
    width: 40px;  
    height: 40px;  
    border-radius: 50%;  
    font-size: 20px;  
    cursor: pointer;  
    display: flex;  
    align-items: center;  
    justify-content: center;  
    transition: all 0.3s;  
}
```

```
.закрыть-рецепт:hover {  
    background: #c0392b;  
    transform: rotate(90deg);  
}
```

```
.рецепт h2 {  
    font-family: 'Cormorant Garamond', serif;  
    font-size: 32px;  
    color: #8b4513;  
    margin-bottom: 30px;  
    text-align: center;  
    display: flex;
```

```
    align-items: center;
    justify-content: center;
    gap: 15px;
}

.рецепт h2 i {
    color: #27ae60;
}

.рецепт-содержимое {
    display: grid;
    grid-template-columns: 1fr 2fr;
    gap: 40px;
}

.ингредиенты, .приготовление {
    background: #f9f5f0;
    padding: 25px;
    border-radius: 10px;
}

.ингредиенты h3, .приготовление h3 {
    font-family: 'Cormorant Garamond', serif;
    font-size: 24px;
    color: #8b4513;
    margin-bottom: 20px;
    display: flex;
    align-items: center;
    gap: 10px;
}

.ингредиенты ul, .приготовление ol {
    padding-left: 25px;
    font-size: 18px;
    color: #5d4037;
    line-height: 1.6;
}

.ингредиенты li, .приготовление li {
    margin-bottom: 10px;
}

.совет-рецепта {
    background: #fff9c4;
```

```
padding: 20px;
border-radius: 10px;
margin-top: 25px;
border-left: 5px solid #f39c12;
font-size: 18px;
color: #7d6608;
}

.совет-рецепта i {
    color: #f39c12;
    margin-right: 10px;
}

/* 12. КНОПКА ДЛЯ РЕЦЕПТА */

.кнопка-рецепта {
    background: linear-gradient(135deg, #d4a762 0%, #b8860b 100%);
    color: white;
    border: none;
    padding: 18px 35px;
    border-radius: 50px;
    font-size: 18px;
    font-weight: 600;
    cursor: pointer;
    display: flex;
    align-items: center;
    justify-content: center;
    gap: 15px;
    margin: 30px auto;
    box-shadow: 0 8px 25px rgba(212, 167, 98, 0.4);
    transition: all 0.3s;
}

.кнопка-рецепта:hover {
    transform: translateY(-5px);
    box-shadow: 0 12px 30px rgba(212, 167, 98, 0.6);
}

/* 13. ПОДСКАЗКА */

.подсказка {
    background: white;
    padding: 25px;
    border-radius: 15px;
    margin-top: 30px;
    max-width: 800px;
```

```
width: 85%;  
border: 2px dashed #3498db;  
}  
  
.подсказка p {  
    font-size: 18px;  
    color: #2c3e50;  
    margin-bottom: 15px;  
    display: flex;  
    align-items: center;  
    gap: 10px;  
}  
  
.подсказка ul {  
    display: grid;  
    grid-template-columns: repeat(auto-fit, minmax(200px, 1fr));  
    gap: 15px;  
    list-style: none;  
    padding-left: 0;  
}  
  
.подсказка li {  
    background: #e8f4fc;  
    padding: 12px 20px;  
    border-radius: 8px;  
    font-size: 16px;  
    color: #2980b9;  
}  
  
/* 14. АДАПТИВНОСТЬ ДЛЯ МОБИЛЬНЫХ */  
@media (max-width: 768px) {  
    body {  
        padding: 15px;  
    }  
  
.письмо {  
    padding: 25px;  
    width: 95vw;  
}  
  
.приветствие h1 {  
    font-size: 36px;  
}
```

```
.основной-текст {
    font-size: 20px;
}

.шапка {
    flex-direction: column;
    gap: 15px;
    align-items: center;
}

.подпись {
    flex-direction: column;
    gap: 15px;
    align-items: center;
    text-align: center;
}

.рецепт-содержимое {
    grid-template-columns: 1fr;
}

.подсказка ul {
    grid-template-columns: 1fr;
}

/*
15. АНИМАЦИЯ ПЕРЕХОДОВ */
@keyframes появление {
    from {
        opacity: 0;
        transform: translateY(20px);
    }
    to {
        opacity: 1;
        transform: translateY(0);
    }
}

.письмо > * {
    animation: появление 0.8s ease-out forwards;
    opacity: 0;
}

.письмо > *:nth-child(1) { animation-delay: 0.1s; }
```

```
.письмо > *:nth-child(2) { animation-delay: 0.2s; }
.письмо > *:nth-child(3) { animation-delay: 0.3s; }
.письмо > *:nth-child(4) { animation-delay: 0.4s; }
.письмо > *:nth-child(5) { animation-delay: 0.5s; }
.письмо > *:nth-child(6) { animation-delay: 0.6s; }
.письмо > *:nth-child(7) { animation-delay: 0.7s; }
.письмо > *:nth-child(8) { animation-delay: 0.8s; }
```

---

## ШАГ 4: Запускаем и восхищаемся!

- Сохраните оба файла в папке ПоздравительноеПисьмо
- Откройте файл index.html в браузере
- Вы увидите красивое анимированное письмо!

**Что можно делать с этим письмом:**

### Эксперимент 1: Меняем цвета

Откройте стили.css и найдите:

css

color: #8b4513; /\* Коричневый \*/

Замените на:

- #2c3e50 — тёмно-синий
- #27ae60 — зелёный
- #8e44ad — фиолетовый
- #c0392b — красный

### Эксперимент 2: Меняем шрифты

В <head> файла письмо.html найдите ссылку Google Fonts и замените шрифты:

- 'Marck Script' → 'Dancing Script'
- 'Cormorant Garamond' → 'Playfair Display'
- 'Open Sans' → 'Roboto'

## Эксперимент 3: Добавляем фото

Вставьте после приветствия:

html

```
<div class="фото-в-письме">
    
    <p class="подпись-фото">Наше foto с тобой в 2010 году</p>
</div>
```

И добавьте в CSS:

css

```
.фото-в-письме {
    text-align: center;
    margin: 30px 0;
}

.фото-в-письме img {
    max-width: 300px;
    border-radius: 10px;
    border: 5px solid #f8e8d0;
    box-shadow: 0 10px 30px rgba(0,0,0,0.1);
}

.подпись-фото {
    font-style: italic;
    color: #666;
    margin-top: 10px;
    font-size: 16px;
}
```

## Эксперимент 4: Делаем интерактивным

Добавьте JavaScript для проигрывания звука:

html

```
<audio id="звукПисьма" src="звук-конверта.mp3" preload="auto"></audio>
<button onclick="document.getElementById('звукПисьма').play()">
     Послушать звук открытия письма
</button>
```

---

## ШАГ 5: Делимся с близкими

### Способ 1: Отправляем ссылку

1. Загрузите папку на Google Диск или Яндекс.Диск
2. Получите ссылку для общего доступа
3. Отправьте ссылку родным

### Способ 2: Распечатываем

1. Нажмите **Ctrl+P** (Печать)
2. В настройках выберите:
  - Цветная печать
  - По размеру страницы
  - Фон страницы (включить)
3. Распечатайте и положите в настоящий конверт!

### Способ 3: Сохраняем как PDF

1. При печати выберите "Сохранить как PDF"
  2. Получится файл, который можно отправить по email
  3. Его можно открыть на любом устройстве
- 

## Чему мы научились в этом практикуме:

- ✓ Создали полноценный веб-проект из двух файлов
- ✓ Применили все изученные свойства CSS:

- `color` и `background-color` — цвета текста и фона
- `font-family` — красивые шрифты из Google Fonts
- `font-size` — разные размеры для разных элементов
- `font-weight` — жирность для акцентов
- `font-style` — курсив для цитат
- `text-align` — выравнивание (left, right, center, justify)

- Добавили интерактивность** — выпадающий рецепт
- Сделали адаптивным** — работает на телефонах и планшетах
- Добавили анимации** — плавное появление элементов
- Использовали сложные CSS-приёмы:**

- Градиенты (linear-gradient)
- Тени (box-shadow)
- Псевдоэлементы (::first-letter)
- Медиа-запросы (@media)
- Flexbox и Grid для вёрстки
- Позиционирование (position)

- Создали настоящий цифровой артефакт**, который можно хранить, печатать, отправлять
- 

## Домашнее задание:

### Задание 1: Создайте письмо для друга

- Измените текст письма (обращение, содержание)
- Подберите другие цвета (синяя гамма для мужчины, розовая для женщины)
- Добавьте реальную фотографию

### Задание 2: Сделайте открытку на 8 Марта

- Уменьшите размеры (ширина 500px)
- Добавьте больше цветочных украшений
- Сделайте кнопку "Отправить виртуальный букет"

### Задание 3: Оформите семейную историю

- Создайте несколько "писем" на одной странице
- Добавьте навигацию между ними
- Вставьте старые семейные фото
- Добавьте голосовые записи воспоминаний

**Бонус:** Создайте "цифровую шкатулку" с коллекцией таких писем для разных случаев:

- Дни рождения

- Поздравления с праздниками
- Благодарственные письма
- Воспоминания о важных событиях

**Поздравляю!** Вы только что создали не просто HTML-страницу, а настоящее цифровое произведение искусства, которое может согреть чье-то сердце. Это и есть магия веб-дизайна — создавать не просто страницы, а эмоции и воспоминания! 

## Часть 2: Осваиваем пространство. Расставляем мебель в комнате

### ● Глава 4: Загадочный БЛОК. Основа основ.

- Что такое блочный элемент? Каждый блок – как отдельная рамка для фотографии.

Дорогой друг, давайте представим, что вы оформляете стену с фотографиями в гостиной. У вас есть разные фотографии, и для каждой вы выбираете рамку. Эта рамка определяет:

- **Сколько места** занимает фотография на стене
- **Расстояние** между этой и соседней фотографией
- **Цвет и толщину** границы вокруг фото
- **Положение** относительно других рамок

В мире веб-дизайна **блочный элемент** — это и есть такая "рамка" для контента! Давайте разберёмся подробно.

---

### 1. Что такое блочный элемент? Простое определение

**Блочный элемент (block element)** — это HTML-элемент, который:

1. **Начинается с новой строки** (как новый абзац в книге)
2. **Занимает всю доступную ширину** родительского контейнера
3. **Может содержать** другие блочные и строчные элементы
4. **Имеет "блочную модель"** — внутренние и внешние отступы

### Примеры блочных элементов (теги HTML):

html

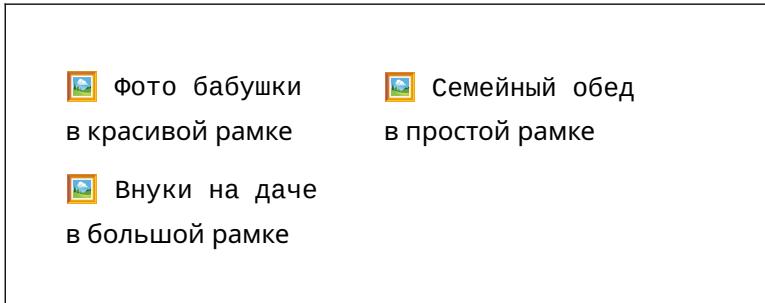
```
<div>      <!-- Универсальный контейнер -->
<h1>-<h6>  <!-- Заголовки -->
<p>        <!-- Абзац -->
<ul>, <ol>  <!-- Списки -->
```

```
<li>      <!-- Элемент списка -->
<section>    <!-- Секция -->
<article>    <!-- Статья -->
<header>    <!-- Шапка -->
<footer>    <!-- Подвал -->
<nav>      <!-- Навигация -->
<main>      <!-- Основное содержимое -->
<aside>    <!-- Боковая панель -->
```

---

## 2. Наглядная аналогия: Стенка с фотографиями

Представьте стену в квартире:



Каждая фотография в рамке — это **блочный элемент**:

- **Фото бабушки** = <div class="фото">
- **Семейный обед** = <div class="фото">
- **Внуки на даче** = <div class="фото большой">

**Особенность:** Каждая рамка начинается с новой строки и занимает всю доступную ширину (пока мы не зададим иначе).

---

## 3. Практический пример: Создаём "стену фотографий"

Создайте файл блочные-элементы.html:

```
html
<!DOCTYPE html>
<html lang="ru">
```

```
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Блочные элементы: Стенка с фотографиями</title>
    <style>
        /* Общие настройки */
        * {
            box-sizing: border-box; /* Важно! Объясню ниже */
        }

        body {
            font-family: 'Arial', sans-serif;
            line-height: 1.6;
            color: #333;
            background-color: #f5f5f5;
            padding: 30px;
        }

        h1 {
            text-align: center;
            color: #2c3e50;
            margin-bottom: 40px;
        }

        /* Раздел 1: Простая аналогия */
        .аналогия {
            background: white;
            padding: 30px;
            border-radius: 15px;
            margin-bottom: 40px;
            box-shadow: 0 5px 15px rgba(0,0,0,0.1);
        }

        .стена-фотографий {
            background: #f8f9fa;
            padding: 30px;
            border-radius: 10px;
            border: 2px dashed #ddd;
            min-height: 400px;
            position: relative;
        }

        /* Рамки-фотографии */
        .рамка {
```

```
background: white;
border: 3px solid;
border-radius: 5px;
padding: 20px;
text-align: center;
font-weight: bold;
color: #2c3e50;
box-shadow: 3px 3px 10px rgba(0,0,0,0.1);
position: absolute; /* Пока абсолютно, потом изменим */
}

.рамка-1 {
    border-color: #e74c3c;
    top: 50px;
    left: 50px;
    width: 200px;
    height: 150px;
}

.рамка-2 {
    border-color: #3498db;
    top: 50px;
    left: 300px;
    width: 250px;
    height: 180px;
}

.рамка-3 {
    border-color: #2ecc71;
    top: 250px;
    left: 50px;
    width: 400px;
    height: 200px;
}

/* Раздел 2: Реальные блочные элементы */
.реальные-примеры {
    background: white;
    padding: 30px;
    border-radius: 15px;
    margin-bottom: 40px;
    box-shadow: 0 5px 15px rgba(0,0,0,0.1);
}
```

```
.пример-блока {  
    background: #f8f9fa;  
    margin: 20px 0;  
    padding: 20px;  
    border-left: 5px solid;  
}  
  
.пример-блока h2 {  
    margin-top: 0;  
    font-size: 18px;  
}  
  
.заголовок-пример { border-left-color: #e74c3c; }  
.абзац-пример { border-left-color: #3498db; }  
.список-пример { border-left-color: #2ecc71; }  
.див-пример { border-left-color: #f39c12; }  
  
/* Розділ 3: Інтерактивна лабораторія */  
.лабораторія {  
    background: white;  
    padding: 30px;  
    border-radius: 15px;  
    margin-bottom: 40px;  
    box-shadow: 0 5px 15px rgba(0,0,0,0.1);  
}  
  
.песочница {  
    background: #f8f9fa;  
    padding: 30px;  
    border-radius: 10px;  
    min-height: 300px;  
    border: 2px solid #e9ecf;  
    margin-top: 20px;  
}  
  
.блок-песочницы {  
    background: #3498db;  
    color: white;  
    padding: 20px;  
    margin: 10px 0;  
    border-radius: 5px;  
    transition: all 0.3s;  
}
```

```
.управление {
    display: grid;
    grid-template-columns: repeat(auto-fit, minmax(200px, 1fr));
    gap: 20px;
    margin-top: 25px;
}

.кнопка {
    background: #2c3e50;
    color: white;
    border: none;
    padding: 15px;
    border-radius: 8px;
    cursor: pointer;
    font-size: 16px;
    transition: background 0.3s;
}

.кнопка:hover {
    background: #3498db;
}

/* Раздел 4: Свойство display */
.display-раздел {
    background: white;
    padding: 30px;
    border-radius: 15px;
    box-shadow: 0 5px 15px rgba(0,0,0,0.1);
}

.сравнение {
    display: flex;
    flex-wrap: wrap;
    gap: 30px;
    margin-top: 25px;
}

.display-пример {
    flex: 1;
    min-width: 250px;
    background: #f8f9fa;
    padding: 25px;
    border-radius: 10px;
}
```

```
.inline-блок {  
    display: inline;  
    background: #e74c3c;  
    color: white;  
    padding: 10px;  
    margin: 5px;  
}  
  
.block-блок {  
    display: block;  
    background: #3498db;  
    color: white;  
    padding: 10px;  
    margin: 5px;  
}  
  
.inline-block-блок {  
    display: inline-block;  
    background: #2ecc71;  
    color: white;  
    padding: 10px;  
    margin: 5px;  
    width: 150px;  
}  
  
/* Информационные блоки */  
.инфо-блок {  
    background: #e8f4fc;  
    padding: 20px;  
    border-radius: 10px;  
    margin: 20px 0;  
    border-left: 5px solid #3498db;  
}  
  
.важное {  
    background: #fff8e1;  
    padding: 20px;  
    border-radius: 10px;  
    margin: 20px 0;  
    border-left: 5px solid #f39c12;  
    color: #7d6608;  
}  
</style>
```

```
</head>
<body>
    <h1> Блочные элементы: Каждый блок — как рамка для фотографии</h1>

    <!-- Раздел 1: Аналогия -->
    <div class="аналогия">
        <h2>1. Аналогия: Стенка с фотографиями в рамках</h2>
        <p>Представьте, что вы вешаете фотографии на стену. Каждая фотография находится в
        своей рамке:</p>

        <div class="стена-фотографий">
            <div class="рамка рамка-1">
                 Фото 1<br>
                (блочный элемент)
            </div>

            <div class="рамка рамка-2">
                 Фото 2<br>
                (другой блок)
            </div>

            <div class="рамка рамка-3">
                 Фото 3<br>
                (большой блок)
            </div>
        </div>

        <div class="инфо-блок">
            <p><strong>Каждая "рамка" (блочный элемент) имеет:</strong></p>
            <ul>
                <li><strong>Собственные границы</strong> — как бортики рамки</li>
                <li><strong>Внутреннее поле</strong> — расстояние от рамки до фото</li>
                <li><strong>Внешние отступы</strong> — расстояние до соседних рамок</li>
                <li><strong>Размер</strong> — ширину и высоту</li>
                <li><strong>Фон</strong> — цвет или рисунок "стенки"</li>
            </ul>
        </div>
    </div>

    <!-- Раздел 2: Реальные примеры -->
    <div class="реальные-примеры">
        <h2>2. Реальные блочные элементы в HTML</h2>
        <p>В HTML многие элементы по умолчанию являются блочными. Вот как они
        выглядят:</p>
    
```

```
<!-- Заголовок -->
<div class="пример-блока заголовок-пример">
    <h2>&lt;h1&gt; - &lt;h6&gt; (Заголовки)</h2>
    <h1 style="margin: 0;">Это заголовок H1 (блочныи элемент)</h1>
    <h2 style="margin: 10px 0 0 0;">Это заголовок H2 (тоже блочныи)</h2>
    <p><strong>Особенность:</strong> Начинаются с новой строки, имеют отступы
сверху и снизу.</p>
</div>

<!-- Абзац -->
<div class="пример-блока абзац-пример">
    <h2>&lt;p&gt; (Абзац)</h2>
    <p style="margin: 0;">Это первый абзац. Он занимает всю доступную ширину.</p>
    <p style="margin: 10px 0 0 0;">Это второй абзац. Он начинается с новой строки.</p>
    <p><strong>Особенность:</strong> Автоматические отступы между абзацами.</p>
</div>

<!-- Список -->
<div class="пример-блока список-пример">
    <h2>&lt;ul&gt; / &lt;ol&gt; (Списки)</h2>
    <ul style="margin: 0; padding-left: 20px;">
        <li>Первый пункт (элемент &lt;li&gt; тоже блочныи!)</li>
        <li>Второй пункт</li>
        <li>Третий пункт</li>
    </ul>
    <p><strong>Особенность:</strong> Весь список — один блочныи элемент, каждый
пункт — тоже блок.</p>
</div>

<!-- Div -->
<div class="пример-блока див-пример">
    <h2>&lt;div&gt; (Универсальный контейнер)</h2>
    <div style="background: #ffeaaf; padding: 15px; margin: 10px 0;">
        Это &lt;div&gt; — самый используемый блочныи элемент. Как пустая рамка,
которую можно наполнить чем угодно!
    </div>
    <p><strong>Особенность:</strong> Не имеет смыслового значения, только
структурное.</p>
    </div>
</div>
```

```
<!-- Раздел 3: Лаборатория -->
<div class="лаборатория">
```

```
<h2>3. Лаборатория: Создайте свои блоки</h2>
```

```
<p>Измените свойства блоков и посмотрите, как они себя ведут:</p>
```

```
<div class="песочница" id="песочница">
  <div class="блок-песочницы" id="блок1">
    Блок 1: Универсальный контейнер
  </div>
```

```
<div class="блок-песочницы" id="блок2">
  Блок 2: Ещё один блок
</div>
```

```
<div class="блок-песочницы" id="блок3">
  Блок 3: И ещё один
</div>
</div>
```

```
<div class="управление">
  <div>
    <h3>Изменяем ширину:</h3>
    <button class="кнопка" onclick="изменитьСвойство('width', '100%')>100%
(полная)</button>
    <button class="кнопка" onclick="изменитьСвойство('width', '50%')>50%
(половина)</button>
    <button class="кнопка" onclick="изменитьСвойство('width', '200px')>200px
(фиксированная)</button>
  </div>
```

```
<div>
  <h3>Изменяем отображение:</h3>
  <button class="кнопка" onclick="изменитьСвойство('display', 'block')>Block
(блочное)</button>
  <button class="кнопка" onclick="изменитьСвойство('display', 'inline')>Inline
(строчное)</button>
  <button class="кнопка" onclick="изменитьСвойство('display', 'inline-block')>Inline-block</button>
</div>
```

```
<div>
  <h3>Другие свойства:</h3>
  <button class="кнопка" onclick="изменитьСвойство('margin', '30px')>Margin:
30px</button>
  <button class="кнопка" onclick="изменитьСвойство('padding', '40px')>Padding:
40px</button>
```

```
<button class="кнопка" onclick="сброситьСтили()">Сбросить</button>
</div>
</div>

<!-- Раздел 4: Display -->
<div class="display-раздел">
    <h2>4. Свойство display: Block vs Inline vs Inline-block</h2>

    <div class="сравнение">
        <div class="display-пример">
            <h3>display: inline (строчные элементы)</h3>
            <p>Строятся в строку, как слова в тексте:</p>
            <div>
                <span class="inline-блок">Элемент 1</span>
                <span class="inline-блок">Элемент 2</span>
                <span class="inline-блок">Элемент 3</span>
            </div>
            <p><strong>Особенности:</strong></p>
            <ul>
                <li>Не начинают новую строку</li>
                <li>Ширина по содержимому</li>
                <li>Нельзя задать width/height</li>
                <li>Вертикальные margin/padding не работают</li>
            </ul>
        </div>
    </div>

    <div class="display-пример">
        <h3>display: block (блочные элементы)</h3>
        <p>Каждый с новой строки, занимают всю ширину:</p>
        <div>
            <div class="block-блок">Элемент 1</div>
            <div class="block-блок">Элемент 2</div>
            <div class="block-блок">Элемент 3</div>
        </div>
        <p><strong>Особенности:</strong></p>
        <ul>
            <li>Начинают новую строку</li>
            <li>Занимают всю доступную ширину</li>
            <li>Можно задать width/height</li>
            <li>Работают все margin/padding</li>
        </ul>
    </div>
</div>
```

```

<div class="display-пример">
    <h3>display: inline-block (гибрид)</h3>
    <p>В строку, но с размерами как у блока:</p>
    <div>
        <div class="inline-block-блок">Элемент 1</div>
        <div class="inline-block-блок">Элемент 2</div>
        <div class="inline-block-блок">Элемент 3</div>
    </div>
    <p><strong>Особенности:</strong></p>
    <ul>
        <li>Не начинают новую строку</li>
        <li>Можно задать width/height</li>
        <li>Работают все margin/padding</li>
        <li>Выравниваются по вертикали</li>
    </ul>
</div>
</div>

<div class="важное">
    <p><strong>💡 Практическое применение:</strong></p>
    <ul>
        <li><strong>&lt;span&gt;, &lt;a&gt;, &lt;strong&gt;</strong> — строчные по
            умолчанию</li>
        <li><strong>&lt;div&gt;, &lt;p&gt;, &lt;h1&gt;</strong> — блочные по
            умолчанию</li>
        <li><strong>&lt;img&gt;, &lt;button&gt;</strong> — inline-block по
            умолчанию</li>
        <li>Можно менять: <code>div { display: inline; }</code> или <code>span { display:
            block; }</code></li>
    </ul>
</div>
</div>

<!-- Шпаргалка -->
<div style="background: #2c3e50; color: white; padding: 30px; border-radius: 15px; margin-top:
40px;">
    <h2 style="color: white; text-align: center;">📋 Шпаргалка: Блочные элементы</h2>

    <div style="display: grid; grid-template-columns: repeat(auto-fit, minmax(300px, 1fr)); gap:
20px; margin-top: 25px;">
        <div style="background: rgba(255,255,255,0.1); padding: 20px; border-radius: 10px;">
            <h3 style="color: white;">🖼 Аналогия рамки</h3>
            <ul>
                <li><strong>Рамка</strong> = Блочный элемент</li>

```

```
<li><strong>Фотография</strong> = Содержимое элемента</li>
<li><strong>Пасспарту</strong> = Padding (внутренний отступ)</li>
<li><strong>Стекло</strong> = Border (граница)</li>
<li><strong>Расстояние до другой рамки</strong> = Margin (внешний
отступ)</li>
</ul>
</div>

<div style="background: rgba(255,255,255,0.1); padding: 20px; border-radius: 10px;">
<h3 style="color: white;"> Основные теги</h3>
<ul>
<li><code>&lt;div&gt;</code> — универсальный контейнер</li>
<li><code>&lt;p&gt;</code> — абзац текста</li>
<li><code>&lt;h1&gt;-&lt;h6&gt;</code> — заголовки</li>
<li><code>&lt;ul&gt;, &lt;ol&gt;</code> — списки</li>
<li><code>&lt;li&gt;</code> — элемент списка</li>
<li><code>&lt;section&gt;</code> — секция страницы</li>
<li><code>&lt;article&gt;</code> — статья</li>
</ul>
</div>

<div style="background: rgba(255,255,255,0.1); padding: 20px; border-radius: 10px;">
<h3 style="color: white;"> Свойства блочных элементов</h3>
<ul>
<li><code>width / height</code> — ширина/высота</li>
<li><code>margin</code> — внешние отступы</li>
<li><code>padding</code> — внутренние отступы</li>
<li><code>border</code> — границы</li>
<li><code>background</code> — фон</li>
<li><code>display</code> — тип отображения</li>
</ul>
</div>
</div>

<div style="text-align: center; margin-top: 30px;">
<div style="background: rgba(255,255,255,0.2); padding: 15px; border-radius: 10px;
display: inline-block;">
<strong>Золотое правило:</strong> Когда нужно создать отдельную "область"
на странице — используйте блочный элемент (обычно &lt;div&gt;).
</div>
</div>
</div>

<script>
```

```
// Лаборатория: изменение свойств блоков
let текущийБлок = 'блок1';

function выбратьБлок(номер) {
    текущийБлок = 'блок' + номер;
    alert('Выбран блок ' + номер + '. Теперь применяйте стили к нему.');
}

function изменитьСвойство(свойство, значение) {
    const блок = document.getElementById(текущийБлок);
    блок.style[свойство] = значение;

    // Показываем подсказку
    блок.title = свойство + ':' + значение;
    блок.style.boxShadow = '0 0 0 3px #3498db';
    setTimeout(() => блок.style.boxShadow = "", 1000);
}

function сброситьСтили() {
    const блоки = ['блок1', 'блок2', 'блок3'];
    блоки.forEach(id => {
        const блок = document.getElementById(id);
        блок.style.cssText = "";
        блок.className = 'блок-песочницы';
        блок.title = "";
    });
    текущийБлок = 'блок1';
}

// Делаем блоки выбираемыми по клику
document.getElementById('блок1').addEventListener('click', () => выбратьБлок(1));
document.getElementById('блок2').addEventListener('click', () => выбратьБлок(2));
document.getElementById('блок3').addEventListener('click', () => выбратьБлок(3));

// Инициализация
document.getElementById('блок1').title = 'Кликните для выбора этого блока';
</script>
</body>
</html>
```

---

## 4. Ключевые особенности блочных элементов

### A. Автоматическое поведение

css

```
/* Блочный элемент по умолчанию: */
div, p, h1 {
    display: block;      /* Блочный тип отображения */
    width: 100%;        /* Занимает всю ширину родителя */
    margin-top: 0;       /* Нет отступа сверху */
    margin-bottom: 1em;  /* Есть отступ снизу */
}
```

### B. Что можно делать с блочными элементами:

#### 1. Задавать размеры:

css

```
.блок {
    width: 300px;      /* Фиксированная ширина */
    height: 200px;     /* Фиксированная высота */
    max-width: 100%;   /* Не шире родителя */
    min-height: 100px; /* Не меньше 100px */
}
```

#### 2. Создавать отступы:

css

```
.блок {
    margin: 20px;      /* Внешние отступы со всех сторон */
    padding: 15px;     /* Внутренние отступы */
    border: 2px solid #ccc; /* Граница */
}
```

#### 3. Располагать на странице:

css

```
.блок {
    float: left;      /* Обтекание слева */
    position: absolute; /* Абсолютное позиционирование */
    display: flex;     /* Flex-контейнер */
}
```

---

## 5. Практическое задание: Создаём фотоальбом

Создайте файл `фотоальбом.html`:

```
html
<!DOCTYPE html>
<html>
<head>
<style>
/* Каждая "рамка" — блочный элемент */
.фото-рамка {
    width: 250px;
    height: 200px;
    margin: 20px;
    padding: 15px;
    border: 10px solid #d4a762;
    background: #ffffaf0;
    display: inline-block; /* Чтобы рамки были в строку */
    box-shadow: 5px 5px 15px rgba(0,0,0,0.2);
    vertical-align: top;
}

/* Заголовок фото — тоже блочный элемент */
.название-фото {
    text-align: center;
    font-weight: bold;
    margin-top: 10px;
    color: #8b4513;
}

/* Контейнер для рамок — блочный элемент */
.фотоальбом {
    width: 90%;
    margin: 0 auto;
    padding: 30px;
    background: #f9f5f0;
    border-radius: 10px;
}

/* Заголовок альбома — блочный элемент */
.заголовок-альбома {
    text-align: center;
    color: #2c3e50;
```

```
border-bottom: 3px solid #d4a762;
padding-bottom: 15px;
margin-bottom: 30px;
}

</style>
</head>
<body>
<div class="фотоальбом">
<h1 class="заголовок-альбома"> Семейный фотоальбом</h1>

<!-- Рамка 1 -->
<div class="фото-рамка">
<div style="width:100%; height:140px; background:#e3f2fd;">
     Фото 1
</div>
<div class="название-фото">Бабушка в саду</div>
</div>

<!-- Рамка 2 -->
<div class="фото-рамка">
<div style="width:100%; height:140px; background:#f3e5f5;">
     Фото 2
</div>
<div class="название-фото">Семейный обед</div>
</div>

<!-- Рамка 3 -->
<div class="фото-рамка">
<div style="width:100%; height:140px; background:#e8f5e9;">
     Фото 3
</div>
<div class="название-фото">Внуки на даче</div>
</div>

<!-- Рамка 4 -->
<div class="фото-рамка">
<div style="width:100%; height:140px; background:#fff3e0;">
     Фото 4
</div>
<div class="название-фото">Новый год</div>
</div>
</div>
</body>
</html>
```

---

## 6. Важное отличие: box-sizing

### Проблема:

css

```
.блок {  
    width: 300px;  
    padding: 20px;  
    border: 5px solid;  
    /* Фактическая ширина = 300 + 20*2 + 5*2 = 350px! */  
}
```

### Решение: box-sizing: border-box

css

```
* {  
    box-sizing: border-box; /* ВАЖНО ДЛЯ ВСЕХ ПЕНСИОНЕРОВ! */  
}  
  
.блок {  
    width: 300px;  
    padding: 20px;  
    border: 5px solid;  
    /* Теперь ширина остаётся 300px! */  
    /* padding и border ВНУТРИ заданной ширины */  
}
```

### Простая аналогия:

- Без border-box: Вы говорите "рамка шириной 30см", а потом добавляете паспарту 5см с каждой стороны → рамка становится 40см.
  - С border-box: Вы говорите "рамка шириной 30см ВКЛЮЧАЯ паспарту" → всё помещается в 30см.
-

## 7. Как определить, блочный ли элемент?

### Способ 1: Посмотреть в браузере

1. Откройте страницу
2. Нажмите F12 (Инструменты разработчика)
3. Наведите курсор на элемент
4. Посмотрите на панель справа — увидите `display: block`

### Способ 2: Простой тест

html

```
<div>Текст 1</div>
<div>Текст 2</div>
```

Если "Текст 2" начинается с новой строки — `<div>` блочный.

### Способ 3: Проверить по умолчанию

Большинство элементов либо блочные, либо строчные:

- **Блочные:** div, p, h1-h6, ul, ol, li, section, article, header, footer, nav, main, aside, form, table
  - **Строчные:** span, a, strong, em, img, input, label, button
- 

## 8. Итог: Простые правила для пенсионеров

### Когда использовать блочные элементы:

1. **Когда нужно создать "область"** на странице (шапка, основная часть, подвал)
2. **Для группировки** связанных элементов
3. **Для создания "рамок"** вокруг контента
4. **Для структурирования** страницы

## Простые аналогии:

- **Рамка для фото** = `<div class="фото-рамка">`
- **Полка для книг** = `<div class="полка">` (в ней уже `<div class="книга">`)
- **Ящик в комоде** = `<div class="ящик">` (в нём `<div class="бельё">`)
- **Этажерка** = несколько `<div class="полка">` друг под другом

## Золотые правила:

1. **Всегда начинайте с:** `* { box-sizing: border-box; }`
2. **Для контейнеров используйте:** `<div>`
3. **Для текста используйте:** `<p>, <h1>-<h6>`
4. **Для списков используйте:** `<ul>/<ol>` с `<li>`
5. **Не бойтесь вкладывать** блоки друг в друга (как матрёшки)

## Простая памятка:

html

```
<!-- Структура типичной страницы -->
<div class="страница">           <!-- Большая рамка (вся страница) -->
    <div class="шапка">...</div>   <!-- Рамка-шапка -->
    <div class="основное">         <!-- Рамка-основное содержимое -->
        <div class="статья">
            <h1>...</h1>          <!-- Рамка-заголовок -->
            <p>...</p>            <!-- Рамка-абзац -->
            <p>...</p>            <!-- Ещё рамка-абзац -->
        </div>
    </div>
    <div class="подвал">...</div>  <!-- Рамка-подвал -->
</div>
```

**Запомните:** Каждый блочный элемент — это отдельная "рамка", которую можно красить, двигать, наполнять и украшать. Это ваш главный строительный блок для создания любых веб-страниц!

Теперь вы понимаете, что такое блочные элементы. В следующей главе мы научимся управлять расстояниями между этими "рамками" — это свойства `margin` и `padding`, которые сделают вашу страницу аккуратной и удобной для чтения! 

■ **Волшебные отступы: margin (расстояние между рамками) и padding (поле внутри рамки, вокруг фото). Объяснение с помощью реальной фоторамки.**

Дорогой друг, представьте, что вы вешаете на стену несколько фотографий в рамках. У вас возникает два важных вопроса:

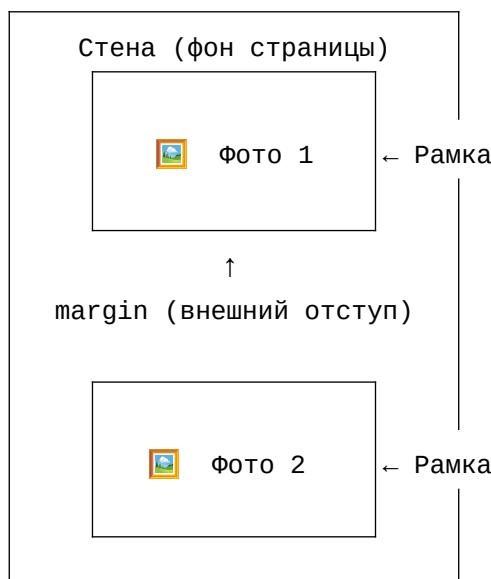
- На каком расстоянии от стены будет сама фотография внутри рамки? (Это padding)**
- На каком расстоянии друг от друга будут висеть разные рамки? (Это margin)**

Давайте разберём эти понятия наглядно и просто!

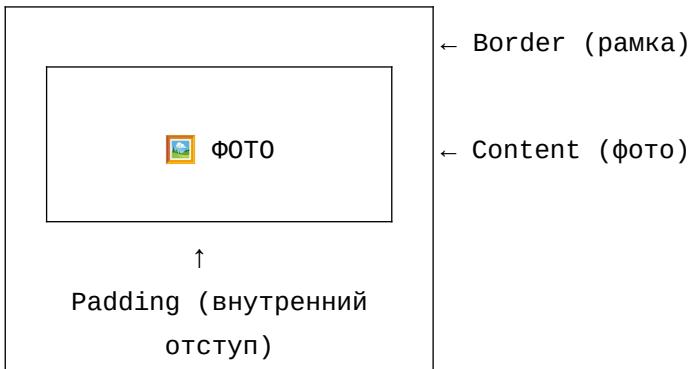
---

## 1. Наглядная аналогия: Стена с фотографиями

Представьте себе такую картину:



А теперь заглянем внутрь одной рамки:



## 2. Padding (падинг) – "Воздух внутри рамки"

### Что такое padding?

**Padding** — это **внутренний отступ** между содержимым элемента и его границей.

**Аналогия:** Расстояние от стекла рамки до самой фотографии. Если padding большой — фотография будет "плавать" внутри рамки. Если маленький — фотография будет вплотную к стеклу.

### Как задаётся padding:

css

```
.рамка {
    padding: 20px; /* Отступ со всех сторон */
}
```

/\* Или более точно: \*/

```
.рамка {
    padding-top: 15px; /* Сверху */
    padding-right: 20px; /* Справа */
    padding-bottom: 15px; /* Снизу */
    padding-left: 20px; /* Слева */
}
```

```
/* Короткая запись (по часовой стрелке): */
.рамка {
    padding: 15px 20px 15px 20px; /* top right bottom left */
}
```

## Наглядный пример padding:

html

```
<div class="рамка с-пэдингом">
    Фото близко к краю
</div>
```

```
<div class="рамка без-пэдинга">
    Фото вплотную к рамке
</div>
```

css

```
.рамка {
    width: 200px;
    height: 150px;
    border: 5px solid brown;
    background: #ffffaf0;
}
```

```
.с-пэдингом {
    padding: 30px; /* Много воздуха внутри */
}
```

```
.без-пэдинга {
    padding: 5px; /* Почти вплотную */
}
```

---

## 3. Margin (марджин) – "Расстояние между рамками"

### Что такое margin?

Margin — это **внешний отступ** между границей элемента и соседними элементами.

**Аналогия:** Расстояние между разными рамками на стене. Если margin большой — рамки висят далеко друг от друга. Если margin маленький или отрицательный — рамки могут соприкасаться или даже накладываться.

## Как задаётся margin:

css

```
.рамка {  
    margin: 30px; /* Отступ со всех сторон */  
}  
  
/* Или более точно:  
 .рамка {  
    margin-top: 20px; /* Сверху */  
    margin-right: 30px; /* Справа */  
    margin-bottom: 20px; /* Снизу */  
    margin-left: 30px; /* Слева */  
}  
  
/* Автоматическое центрирование по горизонтали:  
 .рамка {  
    margin: 0 auto; /* 0 сверху-снизу, auto слева-справа */  
}
```

## Особые значения margin:

css

```
.рамка {  
    margin: 0; /* Без отступов */  
    margin: auto; /* Автоматические отступы (центрирование) */  
    margin: -10px; /* Отрицательный отступ (наложение) */  
}
```

---

## 4. Полный пример: Создаём стену с фотографиями

Создайте файл отступы-пример.html:

html

```
<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Margin и Padding: Стена с фотографиями</title>
    <style>
        /* СБРОС И БАЗОВЫЕ НАСТРОЙКИ */
        * {
            box-sizing: border-box; /* ВАЖНО! */
            margin: 0;
            padding: 0;
        }

        body {
            font-family: 'Arial', sans-serif;
            background: linear-gradient(135deg, #f5f7fa 0%, #c3cfe2 100%);
            min-height: 100vh;
            padding: 30px;
            color: #333;
        }

        h1 {
            text-align: center;
            color: #2c3e50;
            margin-bottom: 40px;
            text-shadow: 1px 1px 2px rgba(0,0,0,0.1);
        }

        /* РАЗДЕЛ 1: НАГЛЯДНАЯ АНАЛОГИЯ */
        .аналогия {
            background: white;
            padding: 30px;
            border-radius: 15px;
            margin-bottom: 40px;
            box-shadow: 0 10px 30px rgba(0,0,0,0.1);
        }

        .визуализация {
            display: flex;
            align-items: center;
            justify-content: center;
            gap: 50px;
            flex-wrap: wrap;
        }
    

```

```
margin-top: 30px;
}

.schema-ramki {
    position: relative;
    width: 300px;
    height: 250px;
    background: #f8f9fa;
    border: 2px dashed #ddd;
    border-radius: 10px;
    padding: 20px;
}

.podpisi {
    background: #f1f8ff;
    padding: 25px;
    border-radius: 10px;
    border: 2px solid #d0e3ff;
    max-width: 400px;
}

.podpis {
    margin: 15px 0;
    padding: 10px 15px;
    border-radius: 5px;
    font-weight: bold;
}

.podpis-mardjin {
    background: #ffea7;
    border-left: 5px solid #fdcb6e;
    color: #7d6608;
}

.podpis-pading {
    background: #d1ecf1;
    border-left: 5px solid #17a2b8;
    color: #0c5460;
}

.podpis-kontent {
    background: #d4edda;
    border-left: 5px solid #28a745;
    color: #155724;
```

```
}

/* РАЗДЕЛ 2: ИНТЕРАКТИВНАЯ СТЕНА */

.интерактивная-стена {
    background: white;
    padding: 30px;
    border-radius: 15px;
    margin-bottom: 40px;
    box-shadow: 0 10px 30px rgba(0,0,0,0.1);
}

.стена {
    background: #f8f9fa;
    padding: 40px;
    border-radius: 10px;
    border: 3px solid #e9ecf;
    min-height: 500px;
    display: flex;
    flex-wrap: wrap;
    gap: 30px;
    justify-content: center;
    align-items: flex-start;
}

.фоторамка {
    background: white;
    border: 8px solid;
    border-radius: 5px;
    width: 200px;
    height: 180px;
    display: flex;
    flex-direction: column;
    align-items: center;
    justify-content: center;
    box-shadow: 5px 5px 15px rgba(0,0,0,0.1);
    transition: all 0.3s ease;
    position: relative;
}

.фоторамка:hover {
    transform: translateY(-5px);
    box-shadow: 8px 8px 20px rgba(0,0,0,0.15);
}
```

```
.фото-область {
    width: 80%;
    height: 60%;
    background: #f0f8ff;
    border: 2px dashed #b3d9ff;
    display: flex;
    align-items: center;
    justify-content: center;
    font-size: 40px;
    color: #4a90e2;
}
```

```
.подпись-фото {
    margin-top: 15px;
    font-weight: bold;
    color: #2c3e50;
}
```

```
/* Цвета для разных рамок */
.рамка-1 { border-color: #e74c3c; }
.рамка-2 { border-color: #3498db; }
.рамка-3 { border-color: #2ecc71; }
.рамка-4 { border-color: #f39c12; }
.рамка-5 { border-color: #9b59b6; }
```

```
/* ПАНЕЛЬ УПРАВЛЕНИЯ */
.панель-управления {
    background: #2c3e50;
    color: white;
    padding: 25px;
    border-radius: 15px;
    margin-top: 30px;
}
```

```
.управление-ряд {
    display: flex;
    flex-wrap: wrap;
    gap: 20px;
    margin: 20px 0;
}
```

```
.селектор-рамки {
    display: flex;
    align-items: center;
```

```
        gap: 10px;
    }

    .кнопка-рамки {
        width: 30px;
        height: 30px;
        border: 3px solid;
        border-radius: 5px;
        cursor: pointer;
        transition: all 0.3s;
    }

    .кнопка-рамки:hover {
        transform: scale(1.1);
    }

    .кнопка-рамки.active {
        box-shadow: 0 0 0 3px white;
    }

    .кн-1 { background: #e74c3c; border-color: #e74c3c; }
    .кн-2 { background: #3498db; border-color: #3498db; }
    .кн-3 { background: #2ecc71; border-color: #2ecc71; }
    .кн-4 { background: #f39c12; border-color: #f39c12; }
    .кн-5 { background: #9b59b6; border-color: #9b59b6; }

    .регуляторы {
        display: grid;
        grid-template-columns: repeat(auto-fit, minmax(200px, 1fr));
        gap: 20px;
    }

    .регулятор {
        background: rgba(255,255,255,0.1);
        padding: 20px;
        border-radius: 10px;
    }

    .ползунок {
        width: 100%;
        margin: 15px 0;
    }

    .кнопка-действия {
```

```
background: #3498db;
color: white;
border: none;
padding: 12px 24px;
margin: 5px;
border-radius: 6px;
cursor: pointer;
font-size: 16px;
transition: all 0.3s;
}

.кнопка-действия:hover {
background: #2980b9;
transform: translateY(-2px);
}

.значение {
font-family: 'Courier New', monospace;
background: rgba(0,0,0,0.2);
padding: 5px 10px;
border-radius: 4px;
margin-left: 10px;
}

/* РАЗДЕЛ 3: РЕАЛЬНЫЕ ПРИМЕРЫ */
.реальные-примеры {
background: white;
padding: 30px;
border-radius: 15px;
margin-bottom: 40px;
box-shadow: 0 10px 30px rgba(0,0,0,0.1);
}

.примеры-ряд {
display: grid;
grid-template-columns: repeat(auto-fit, minmax(300px, 1fr));
gap: 25px;
margin-top: 30px;
}

.пример-карточка {
background: #f8f9fa;
padding: 25px;
border-radius: 10px;
```

```
border: 2px solid #e9ecef;
}

.визуальный-пример {
    background: white;
    padding: 20px;
    border-radius: 8px;
    margin: 20px 0;
    border: 1px solid #ddd;
    position: relative;
}

.код-пример {
    background: #2c3e50;
    color: white;
    padding: 15px;
    border-radius: 8px;
    font-family: 'Courier New', monospace;
    font-size: 14px;
    margin-top: 15px;
}

/* ИНФО-БЛОКИ */
.инфо-блок {
    background: #e8f4fc;
    padding: 20px;
    border-radius: 10px;
    margin: 20px 0;
    border-left: 5px solid #3498db;
}

.совет {
    background: #fff8e1;
    padding: 20px;
    border-radius: 10px;
    margin: 20px 0;
    border-left: 5px solid #f39c12;
}

/* ПОДСКАЗКИ НА САМИХ РАМКАХ */
.подсказка-марджин {
    position: absolute;
    background: #ffea7;
    color: #7d6608;
```

```

padding: 5px 10px;
border-radius: 4px;
font-size: 12px;
font-weight: bold;
pointer-events: none;
z-index: 10;
}

.подсказка-падинг {
position: absolute;
background: #d1ecf1;
color: #0c5460;
padding: 5px 10px;
border-radius: 4px;
font-size: 12px;
font-weight: bold;
pointer-events: none;
z-index: 10;
}

.подсказка-марджин::before, .подсказка-падинг::before {
content: "↑ ";
}

</style>
</head>
<body>
<h1> Margin и Padding: Искусство правильных расстояний</h1>

<!-- РАЗДЕЛ 1: АНАЛОГИЯ -->
<div class="аналогия">
<h2>1. Наглядная аналогия: Фотография в рамке на стене</h2>

<div class="визуализация">
<div class="схема-рамки">
<!-- Визуализация отступов -->
<div style="position: absolute; top: -25px; left: 0; right: 0; text-align: center;">
<div class="подсказка-марджин" style="background: #ffeaa7;">margin-
top</div>
</div>

<div style="position: absolute; left: -80px; top: 50%; transform: translateY(-50%);">
<div class="подсказка-марджин" style="transform: rotate(-90deg);">margin-
left</div>
</div>

```

```
<!-- Border (рамка) -->
<div style="width: 100%; height: 100%; border: 15px solid #d4a762; border-radius: 3px; position: relative;">

    <!-- Padding (наснапмы) -->
    <div style="position: absolute; top: 20px; left: 20px; right: 20px; bottom: 20px; background: #ffffaf0; border: 2px dashed #d4a762; display: flex; align-items: center; justify-content: center;">

        <!-- Content (фото) -->
        <div style="width: 80%; height: 60%; background: #4a90e2; display: flex; align-items: center; justify-content: center; color: white; font-weight: bold; font-size: 20px;">
             ФОТО
        </div>

        <!-- Подсказка padding -->
        <div class="подсказка-падинг" style="top: -15px; left: 50%; transform: translateX(-50%); padding">
            </div>
        </div>

        <!-- Подсказки margin справа и снизу -->
        <div style="position: absolute; right: -80px; top: 50%; transform: translateY(-50%);">
            <div class="подсказка-марджин" style="transform: rotate(90deg);>margin-right</div>
        </div>

        <div style="position: absolute; bottom: -25px; left: 0; right: 0; text-align: center;">
            <div class="подсказка-марджин">margin-bottom</div>
        </div>
        </div>

<div class="подписи">
    <h3>Что есть что:</h3>

    <div class="подпись подпись-марджин">
        <strong>MARGIN</strong> — внешний отступ<br>
        <small>Расстояние ОТ этой рамки ДО других рамок или края стены</small>
    </div>

    <div class="подпись подпись-падинг">
        <strong>PADDING</strong> — внутренний отступ<br>
        <small>Расстояние ОТ края рамки ДО самой фотографии внутри</small>
    </div>
</div>
```

```
</div>

<div class="подпись подпись-контент">
    <strong>CONTENT</strong> — содержимое<br>
    <small>Сама фотография, текст или другой контент</small>
</div>

<div class="совет" style="margin-top: 20px;">
    <strong>💡 Простое правило:</strong><br>
    <strong>Margin</strong> = "Не трогайте меня, мне нужно пространство
вокруг!"<br>
    <strong>Padding</strong> = "Моему содержимому нужно дышать внутри
меня!"<br>
</div>
</div>
</div>

<!-- РАЗДЕЛ 2: ИНТЕРАКТИВНАЯ СТЕНА -->
<div class="интерактивная-стена">
    <h2>2. Интерактивная стена: Поиграйте с отступами!</h2>
    <p>Выберите рамку и регулируйте её margin и padding. Смотрите, как меняются
расстояния.</p>

<div class="стена" id="стена">
    <!-- Рамка 1 -->
    <div class="фоторамка рамка-1" id="рамка1" data-index="1">
        <div class="фото-область">
            
        </div>
        <div class="подпись-фото">Фото 1</div>
        <div class="подсказка-марджин" style="top: -25px;">margin</div>
        <div class="подсказка-падинг" style="bottom: -25px;">padding</div>
    </div>

    <!-- Рамка 2 -->
    <div class="фоторамка рамка-2" id="рамка2" data-index="2">
        <div class="фото-область">
            
        </div>
        <div class="подпись-фото">Фото 2</div>
        <div class="подсказка-марджин" style="top: -25px;">margin</div>
        <div class="подсказка-падинг" style="bottom: -25px;">padding</div>
    </div>
</div>
```

```

<!-- Рамка 3 -->

    </div>
    <div class="подпись-фото">Фото 3</div>
    <div class="подсказка-марджин" style="top: -25px;">margin</div>
    <div class="подсказка-падинг" style="bottom: -25px;">padding</div>
</div>

<!-- Рамка 4 -->

    </div>
    <div class="подпись-фото">Фото 4</div>
    <div class="подсказка-марджин" style="top: -25px;">margin</div>
    <div class="подсказка-падинг" style="bottom: -25px;">padding</div>
</div>

<!-- Рамка 5 -->

    </div>
    <div class="подпись-фото">Фото 5</div>
    <div class="подсказка-марджин" style="top: -25px;">margin</div>
    <div class="подсказка-падинг" style="bottom: -25px;">padding</div>
</div>
</div>

<!-- ПАНЕЛЬ УПРАВЛЕНИЯ -->

```

```
</div>

<div class="регуляторы">
    <div class="регулятор">
        <h4 style="color: white; margin-bottom: 15px;">Margin (внешний
отступ)</h4>

        <label>
            Сверху: <span id="marginTopValue" class="значение">20px</span>
            <input type="range" class="ползунок" id="marginTop" min="0"
max="100" value="20" oninput="изменитьMargin('top', this.value)">
        </label>

        <label>
            Справа: <span id="marginRightValue" class="значение">20px</span>
            <input type="range" class="ползунок" id="marginRight" min="0"
max="100" value="20" oninput="изменитьMargin('right', this.value)">
        </label>

        <label>
            Снизу: <span id="marginBottomValue" class="значение">20px</span>
            <input type="range" class="ползунок" id="marginBottom" min="0"
max="100" value="20" oninput="изменитьMargin('bottom', this.value)">
        </label>

        <label>
            Слева: <span id="marginLeftValue" class="значение">20px</span>
            <input type="range" class="ползунок" id="marginLeft" min="0"
max="100" value="20" oninput="изменитьMargin('left', this.value)">
        </label>

        <div style="margin-top: 15px;">
            <button class="кнопка-действия"
onclick="установитьMargin('0')">0px</button>
            <button class="кнопка-действия"
onclick="установитьMargin('20')">20px</button>
            <button class="кнопка-действия"
onclick="установитьMargin('50')">50px</button>
        </div>
    </div>

    <div class="регулятор">
        <h4 style="color: white; margin-bottom: 15px;">Padding (внутренний
отступ)</h4>
```

```
<label>
    Сверху: <span id="paddingTopValue" class="значение">15px</span>
    <input type="range" class="ползунок" id="paddingTop" min="0"
max="50" value="15" oninput="изменитьPadding('top', this.value)">
</label>

<label>
    Справа: <span id="paddingRightValue" class="значение">15px</span>
    <input type="range" class="ползунок" id="paddingRight" min="0"
max="50" value="15" oninput="изменитьPadding('right', this.value)">
</label>

<label>
    Снизу: <span id="paddingBottomValue" class="значение">15px</span>
    <input type="range" class="ползунок" id="paddingBottom" min="0"
max="50" value="15" oninput="изменитьPadding('bottom', this.value)">
</label>

<label>
    Слева: <span id="paddingLeftValue" class="значение">15px</span>
    <input type="range" class="ползунок" id="paddingLeft" min="0"
max="50" value="15" oninput="изменитьPadding('left', this.value)">
</label>

<div style="margin-top: 15px;">
    <button class="кнопка-действия"
onclick="установитьPadding('5')">5px</button>
    <button class="кнопка-действия"
onclick="установитьPadding('15')">15px</button>
    <button class="кнопка-действия"
onclick="установитьPadding('30')">30px</button>
</div>
</div>

<div style="text-align: center; margin-top: 25px;">
    <button class="кнопка-действия" onclick="сброситьВсе()" style="background:
#e74c3c;">Сбросить все</button>
    <button class="кнопка-действия" onclick="показатьКод()" style="background:
#2ecc71;">Показать CSS-код</button>
</div>
</div>
```

```
<!-- РАЗДЕЛ 3: РЕАЛЬНЫЕ ПРИМЕРЫ -->
<div class="реальные-примеры">
    <h2>3. Практические примеры использования</h2>

    <div class="примеры-ряд">
        <!-- Пример 1: Карточка -->
        <div class="пример-карточка">
            <h3>Карточка товара</h3>
            <div class="визуальный-пример" style="padding: 0; overflow: hidden; border-radius: 10px;">
                <div style="background: #3498db; padding: 20px; text-align: center; color: white;">
                     Смартфон
                </div>
                <div style="padding: 20px;">
                    <p>Отличный телефон для пенсионеров. Большие кнопки, громкий звук.</p>
                    <div style="margin-top: 15px; padding: 15px; background: #f8f9fa; border-radius: 5px;">
                        Цена: <strong>5 000 руб.</strong>
                    </div>
                </div>
            </div>
            <div class="код-пример">
                .карточка {<br>
                    &nbsp;&nbsp; margin: 20px;&nbsp;&nbsp;<span style="color: #ffea7;">/*
                Отступ от других карточек */</span><br>
                    &nbsp;&nbsp; padding: 0;&nbsp;&nbsp;&nbsp;&nbsp;<span style="color: #d1ecf1;">/*
                Нет внутренних отступов у самой карточки */</span><br>
                }<br><br>
                .карточка-заголовок {<br>
                    &nbsp;&nbsp; padding: 20px;&nbsp;&nbsp;<span style="color: #d1ecf1;">/*
                Воздух вокруг текста в заголовке */</span><br>
                }<br><br>
                .карточка-тело {<br>
                    &nbsp;&nbsp; padding: 20px;&nbsp;&nbsp;<span style="color: #d1ecf1;">/*
                Воздух вокруг основного текста */</span><br>
                }
            </div>
        </div>
        <!-- Пример 2: Кнопка -->
        <div class="пример-карточка">
```

```
<h3>Кнопка</h3>
<div class="визуальный-пример" style="text-align: center;">
    <button style="padding: 15px 30px; background: #2ecc71; color: white; border:
none; border-radius: 5px; font-size: 18px; margin: 10px;">
        Купить сейчас
    </button>
    <div style="margin-top: 20px;">
        <button style="padding: 8px 16px; margin: 5px;">Маленькая</button>
        <button style="padding: 12px 24px; margin: 5px;">Средняя</button>
        <button style="padding: 16px 32px; margin: 5px;">Большая</button>
    </div>
</div>
<div class="код-пример">
    .кнопка {<br>
        &nbsp;&nbsp; padding: 15px 30px;&nbsp;&nbsp;<span style="color:
#d1ecf1;"> /* 15px сверху-снизу, 30px слева-справа */</span><br>
        &nbsp;&nbsp; margin: 10px;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<span
style="color: #ffeaaf;"> /* Отступ между кнопками */</span><br>
    }
</div>
<div class="совет" style="margin-top: 15px; padding: 10px; font-size: 14px;">
    <strong>💡 Совет:</strong> Чем больше padding у кнопки — тем она
кажется важнее и удобнее для нажатия.
</div>
</div>
```

```
<!-- Пример 3: Абзац текста -->
<div class="пример-карточка">
    <h3>Текст для чтения</h3>
    <div class="визуальный-пример">
        <h4 style="margin-bottom: 15px;">Заголовок статьи</h4>
        <p style="margin-bottom: 15px; line-height: 1.6;">
            Это первый абзац статьи. У него есть отступ снизу, чтобы отделиться
от следующего абзаца.
    </p>
    <p style="line-height: 1.6;">
        Это второй абзац. Благодаря margin между абзацами текст легко
читается и не сливаются.
```

```
</p>
</div>
<div class="код-пример">
    h4 {<br>
        &nbsp;&nbsp; margin-bottom: 15px;&nbsp;&nbsp;<span style="color:
#ffeaaf;"> /* Отступ от заголовка до текста */</span><br>
```

```
 }<br><br>
 p {<br>
   &nbsp;&nbsp;margin-bottom: 15px;&nbsp;&nbsp;<span style="color: #ffea7;">/* Отступ между абзацами */</span><br>
   &nbsp;&nbsp;line-height: 1.6;&nbsp;&nbsp;&nbsp;&nbsp;&nbsp;<span style="color: #d1ecf1;">/* Межстрочный интервал */</span><br>
 }
</div>
</div>
</div>
```

```
<!-- ШПАРГАЛКА -->
<div style="background: linear-gradient(135deg, #667eea 0%, #764ba2 100%); color: white; padding: 30px; border-radius: 15px;">
  <h2 style="color: white; text-align: center;">📋 Шпаргалка: Margin vs Padding</h2>
```

```
  <div style="display: grid; grid-template-columns: repeat(auto-fit, minmax(250px, 1fr)); gap: 20px; margin-top: 25px;">
    <div style="background: rgba(255,255,255,0.1); padding: 20px; border-radius: 10px; backdrop-filter: blur(10px);">
      <h3 style="color: white;">MARGIN</h3>
      <ul>
        <li><strong>Внешний отступ</strong></li>
        <li>Между разными элементами</li>
        <li>Может быть отрицательным</li>
        <li>Не имеет фона</li>
        <li>Схлопывается с соседними margin</li>
      </ul>
      <div style="margin-top: 15px; padding: 10px; background: rgba(255,234,167,0.3); border-radius: 5px;">
        <strong>Как запомнить:</strong> Margin = Мое Личное Пространство
      </div>
    </div>
```

```
  <div style="background: rgba(255,255,255,0.1); padding: 20px; border-radius: 10px; backdrop-filter: blur(10px);">
    <h3 style="color: white;">PADDING</h3>
    <ul>
      <li><strong>Внутренний отступ</strong></li>
      <li>Внутри элемента, до его границы</li>
      <li>Всегда положительный</li>
      <li>Имеет фон элемента</li>
      <li>Не схлопывается</li>
    </ul>
```

```
</ul>
<div style="margin-top: 15px; padding: 10px; background: rgba(209,236,241,0.3); border-radius: 5px;">
    <strong>Как запомнить:</strong> Padding = Подушка для контента
</div>
</div>

<div style="background: rgba(255,255,255,0.1); padding: 20px; border-radius: 10px; backdrop-filter: blur(10px);">
    <h3 style="color: white;">КОГДА ЧТО ИСПОЛЬЗОВАТЬ?</h3>
    <ul>
        <li><strong>Margin</strong> — для расстояния между блоками</li>
        <li><strong>Padding</strong> — для воздуха внутри блока</li>
        <li><strong>Margin: auto</strong> — для центрирования</li>
        <li><strong>Padding</strong> — если нужен фон у отступа</li>
        <li><strong>Margin</strong> — для наложения элементов</li>
    </ul>
</div>
</div>

<div style="text-align: center; margin-top: 30px;">
    <div style="background: rgba(255,255,255,0.2); padding: 15px; border-radius: 10px; display: inline-block;">
        <strong>💡 Золотое правило:</strong> Margin — наружу, Padding — внутрь.
        Margin отделяет элемент от других, Padding отделяет контент от границ элемента.
    </div>
    </div>
</div>

<script>
    // ПЕРЕМЕННЫЕ ДЛЯ УПРАВЛЕНИЯ
    let выбраннаяРамка = 1;

    // ВЫБОР РАМКИ
    function выбратьРамку(номер) {
        выбраннаяРамка = номер;

        // Обновляем активную кнопку
        document.querySelectorAll('.кнопка-рамки').forEach(btn => {
            btn.classList.remove('active');
        });
        document.querySelector(`.кн-${номер}`).classList.add('active');
    }
</script>
```

```
// Подсвечиваем выбранную рамку
document.querySelectorAll('.фоторамка').forEach(рамка => {
    рамка.style.boxShadow = '5px 5px 15px rgba(0,0,0,0.1)';
});
document.getElementById(`рамка${номер}`).style.boxShadow = '0 0 0 3px white, 5px 5px
15px rgba(0,0,0,0.2)';

// Обновляем значения ползунков
const рамка = document.getElementById(`рамка${номер}`);
const стили = window.getComputedStyle(рамка);

// Margin значения
document.getElementById('marginTop').value = parseInt(стили.marginTop) || 0;
document.getElementById('marginTopValue').textContent = стили.marginTop;

document.getElementById('marginRight').value = parseInt(стили.marginRight) || 0;
document.getElementById('marginRightValue').textContent = стили.marginRight;

document.getElementById('marginBottom').value = parseInt(стили.marginBottom) || 0;
document.getElementById('marginBottomValue').textContent = стили.marginBottom;

document.getElementById('marginLeft').value = parseInt(стили.marginLeft) || 0;
document.getElementById('marginLeftValue').textContent = стили.marginLeft;

// Padding значения
document.getElementById('paddingTop').value = parseInt(стили.paddingTop) || 0;
document.getElementById('paddingTopValue').textContent = стили.paddingTop;

document.getElementById('paddingRight').value = parseInt(стили.paddingRight) || 0;
document.getElementById('paddingRightValue').textContent = стили.paddingRight;

document.getElementById('paddingBottom').value = parseInt(стили.paddingBottom) ||
0;
document.getElementById('paddingBottomValue').textContent = стили.paddingBottom;

document.getElementById('paddingLeft').value = parseInt(стили.paddingLeft) || 0;
document.getElementById('paddingLeftValue').textContent = стили.paddingLeft;
}

// ИЗМЕНЕНИЕ MARGIN
function изменитьMargin(сторона, значение) {
    const рамка = document.getElementById(`рамка${выбраннаяРамка}`);
    рамка.style[`margin${сторона.charAt(0).toUpperCase() + сторона.slice(1)}`] = `$
{значение}px`;
```

```

// Обновляем отображение значения
document.getElementById(`margin${сторона.charAt(0).toUpperCase() +
сторона.slice(1)}Value`).textContent = `${значение}px`;
}

function установитьMargin(значение) {
    const рамка = document.getElementById(`рамка${выбраннаяРамка}`);
    рамка.style.margin = `${значение}px`;

    // Обновляем все значения
    ['Top', 'Right', 'Bottom', 'Left'].forEach(сторона => {
        document.getElementById(`margin${сторона}`).value = значение;
        document.getElementById(`margin${сторона}Value`).textContent = ` ${
значение}px`;
    });
}

// ИЗМЕНЕНИЕ PADDING
function изменитьPadding(сторона, значение) {
    const рамка = document.getElementById(`рамка${выбраннаяРамка}`);
    рамка.style[`padding${сторона.charAt(0).toUpperCase() + сторона.slice(1)}] = ` ${
значение}px`;

    // Обновляем отображение значения
    document.getElementById(`padding${сторона.charAt(0).toUpperCase() + сторона.slice(1)}Value`).textContent = `${значение}px`;
}

function установитьPadding(значение) {
    const рамка = document.getElementById(`рамка${выбраннаяРамка}`);
    рамка.style.padding = `${значение}px`;

    // Обновляем все значения
    ['Top', 'Right', 'Bottom', 'Left'].forEach(сторона => {
        document.getElementById(`padding${сторона}`).value = значение;
        document.getElementById(`padding${сторона}Value`).textContent = ` ${
значение}px`;
    });
}

// СБРОС ВСЕХ СТИЛЕЙ
function сброситьВсе() {
    for (let i = 1; i <= 5; i++) {

```

```
const рамка = document.getElementById(`рамка${i}`);
рамка.style.margin = '20px';
рамка.style.padding = '15px';
}

// Сбрасываем ползунки
выбратьРамку(выбраннаяРамка);
}

// ПОКАЗАТЬ CSS-КОД
function показатьКод() {
    const рамка = document.getElementById(`рамка${выбраннаяРамка}`);
    const стили = window.getComputedStyle(рамка);

    const код = `#рамка${выбраннаяРамка} {\n` +
        ` margin: ${стили.marginTop} ${стили.marginRight} $ ${стили.marginBottom} ${стили.marginLeft};\n` +
        ` padding: ${стили.paddingTop} ${стили.paddingRight} $ ${стили.paddingBottom} ${стили.paddingLeft};\n` +
        `}`;

    alert(`CSS-код для рамки ${выбраннаяРамка}:\n\n${код}\n\nСкопируйте и используйте в своих проектах!`);
}

// ИНИЦИАЛИЗАЦИЯ
window.onload = function() {
    выбратьРамку(1);

    // Делаем все рамки кликабельными
    for (let i = 1; i <= 5; i++) {
        document.getElementById(`рамка${i}`).addEventListener('click', function(e) {
            if (e.target === this || this.contains(e.target)) {
                выбратьРамку(i);
            }
        });
    }
};

</script>
</body>
</html>
```

---

## 5. Практические примеры для пенсионеров

### Пример 1: Визитная карточка

html

```
<div class="визитка">
    <h2>Галина Ивановна</h2>
    <p>Садовод-любитель с 40-летним стажем</p>
    <p>г. Москва, ул. Садовая, 15</p>
    <p>тел.: 8-999-123-45-67</p>
</div>
```

css

```
.визитка {
    width: 300px;
    padding: 30px;      /* Воздух внутри визитки */
    margin: 20px auto;  /* Центрирование + отступы сверху/снизу */
    border: 2px solid gold; /* Золотая рамка */
    background: white;
    text-align: center;
    border-radius: 10px;
    box-shadow: 0 5px 15px rgba(0,0,0,0.1);
}
```

```
.визитка h2 {
    margin-bottom: 15px; /* Отступ от имени до профессии */
    color: #8b4513;
}
```

```
.визитка p {
    margin: 8px 0;      /* Отступы между строками */
    color: #666;
}
```

### Пример 2: Меню для дачи

html

```
<div class="меню-дача">
    <h3>Что сделать на даче сегодня:</h3>
    <ul>
        <li>Полить помидоры</li>
        <li>Прополоть грядки</li>
```

```

<li>Собрать ягоды</li>
<li>Покосить траву</li>
</ul>
</div>
css

.меню-дача {
    padding: 20px;          /* Воздух внутри блока */
    margin: 15px;           /* Отступ от других элементов */
    background: #f0ffff;    /* Светло-зелёный фон */
    border-left: 5px solid #2ecc71; /* Зелёная полоска слева */
}

.меню-дача h3 {
    margin-top: 0;          /* Убираем отступ сверху */
    margin-bottom: 15px;     /* Отступ до списка */
    color: #27ae60;
}

.меню-дача ul {
    margin: 0;              /* Убираем отступы у списка */
    padding-left: 20px;      /* Отступ для маркеров списка */
}

.меню-дача li {
    margin-bottom: 8px;      /* Отступ между пунктами */
    padding: 5px 0;          /* Воздух вокруг текста пункта */
}

```

---

## 6. Важные особенности и советы

### A. Схлопывание margin (Collapsing Margins)

css

```

/* Если два элемента имеют margin: */
.блок1 { margin-bottom: 30px; }
.блок2 { margin-top: 20px; }

/* Между ними будет НЕ 50px, а 30px (берется больший margin)! */

```

**Решение:**

- Используйте только один margin (или сверху, или снизу)
- Добавьте padding или border между элементами
- Используйте Flexbox или Grid layout

## B. Отрицательные margin

css

```
.блок {  
    margin: -10px; /* Может выходить за родителя! */  
}
```

**Используйте осторожно!** Может создавать наложения.

## C. Автоматические margin для центрирования

css

```
.блок {  
    width: 500px;  
    margin: 0 auto; /* Центрирование по горизонтали */  
    /* 0 - сверху/снизу, auto - слева/справа */  
}
```

## D. Разные способы записи

css

```
/* Полная запись */  
.блок {  
    margin-top: 10px;  
    margin-right: 20px;  
    margin-bottom: 10px;  
    margin-left: 20px;  
}  
  
/* Сокращённая запись (по часовой стрелке) */  
.блок {  
    margin: 10px 20px 10px 20px; /* top right bottom left */  
}
```

```
/* Ещё короче (top/bottom и right/left одинаковы) */  
.блок {
```

```
margin: 10px 20px; /* 10px сверху/снизу, 20px слева/справа */  
}  
  
/* Все стороны одинаковы */  
.блок {  
    margin: 15px; /* Все 4 стороны по 15px */  
}
```

---

## 7. Итог: Простые правила для пенсионеров

### Когда использовать MARGIN:

1. Чтобы отодвинуть рамку от другой рамки
2. Чтобы центрировать блок (margin: 0 auto;)
3. Чтобы создать пространство между разными элементами
4. Когда не нужен фон у отступа

### Когда использовать PADDING:

1. Чтобы отодвинуть фото от края рамки
2. Чтобы текст не прилипал к границам блока
3. Когда нужен фон у отступа (у padding есть фон элемента)
4. Чтобы увеличить кликабельную область кнопки

### Простая памятка:

```
css  
  
/* Для кнопок */  
.кнопка {  
    padding: 10px 20px; /* Воздух внутри кнопки */  
    margin: 5px; /* Расстояние между кнопками */  
}  
  
/* Для карточек */  
.карточка {  
    padding: 20px; /* Воздух внутри карточки */  
    margin: 15px; /* Расстояние между карточками */  
}
```

```
/* Для текста */
.абзац {
    padding: 0;          /* Текст вплотную к границам */
    margin: 15px 0;      /* Отступы между абзацами */
}

/* Для центрирования */
.центрированный-блок {
    width: 500px;
    padding: 20px;       /* Воздух внутри */
    margin: 0 auto;      /* Центрирование по горизонтали */
}
```

## Золотые правила:

1. **Начинайте с** `* { box-sizing: border-box; }` — это избавит от головной боли с размерами
2. **Используйте padding для "воздуха" внутри**, margin — для расстояния между элементами
3. **Экспериментируйте** с разными значениями в инспекторе браузера (F12)
4. **Проверяйте на мобильных** — большие отступы могут съесть весь экран
5. **Будьте последовательны** — используйте одинаковые отступы для похожих элементов

## Простая аналогия для запоминания:

- **MARGIN** = Мое Личное Пространство (не подходи близко!)
- **PADDING** = Подушка для контента (чтобы ему было удобно внутри)

**Теперь вы мастер отступов!** Вы можете создавать аккуратные, сбалансированные интерфейсы, где каждый элемент имеет своё законное пространство. В следующей главе мы научимся создавать красивые рамки и тени — это сделает ваши "фотографии" ещё более эффектными!  

## ■ Практикум №3: Создаем свою первую визитку или открытку с отступами.

Дорогой друг, давайте создадим что-то реальное и полезное — цифровую визитку или открытку, которую можно отправить друзьям, распечатать или сохранить на память! Мы применим всё, что узнали об отступах (margin и padding), и создадим настоящее произведение веб-дизайна.

### ШАГ 1: Выбираем проект — Визитка ИЛИ Открытка

#### Вариант А: Визитная карточка

Идеально для:

- Представления себя как мастера/рукодельницы/садовода
- Контактов для соседей по даче
- Обмена информацией в клубе по интересам

#### Вариант Б: Поздравительная открытка

Идеально для:

- Дней рождения родных
- Праздников (8 Марта, 23 Февраля, Новый год)
- Просто "привет, я о тебе помню!"

---

### ШАГ 2: Подготовка рабочего пространства

#### Создайте папку и файлы:

1. На Рабочем столе создайте папку **МояПерваяВизитка** (или **МояПерваяОткрытка**)
2. Внутри создайте:

- index.html — основной файл
  - style.css — файл стилей
  - папку фото — для ваших фотографий (можно добавить позже)
- 

## ШАГ 3: Создаём БАЗОВЫЙ ШАБЛОН (общий для всех)

### Файл index.html:

```
html

<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Моя первая визитка | CSS для пенсионеров</title>

    <!-- Подключаем красивые шрифты -->
    <link rel="preconnect" href="https://fonts.googleapis.com">
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
    <link href="https://fonts.googleapis.com/css2?
family=Cormorant+Garamond:wght@400;600&family=Marck+Script&family=Open+Sans:wght@300;400&
display=swap" rel="stylesheet">

    <!-- Подключаем иконки -->
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/
all.min.css">

    <!-- Подключаем наши стили -->
    <link rel="stylesheet" href="style.css">

    <!-- Фавиконка (маленькая иконка во вкладке) -->
    <link rel="icon" href="data:image/svg+xml,<svg xmlns=%22http://www.w3.org/2000/svg%22
viewBox=%220 0 100 100%22><text y=%22.9em%22 font-size=%2290%22>🍪</text></svg>">
</head>
<body>
    <!-- Контейнер для выбора проекта -->
    <div class="выбор-проекта">
        <h1>🍪 Мой первый веб-проект</h1>
        <p>Выберите, что хотите создать:</p>
        <div class="кнопки-выбора">
```

```
<button class="кнопка-проекта активная" data-проект="визитка"> Визитная  
карточка</button>  
    <button class="кнопка-проекта" data-проект="открытка"> Поздравительная  
открытка</button>  
    </div>  
</div>  
  
<!-- Основной контейнер, куда будет загружаться выбранный проект -->  
<div class="контейнер" id="контейнерПроекта">  
    <!-- Сюда будет загружаться выбранный проект -->  
</div>  
  
<!-- Панель управления отступами -->  
<div class="панель-управления">  
    <h3><i class="fas fa-sliders-h"></i> Лаборатория отступов</h3>  
    <p>Измените значения и посмотрите, как меняется ваша визитка/открытка!</p>  
  
<div class="регуляторы">  
    <div class="регулятор-группа">  
        <h4>PADDING (внутренние отступы)</h4>  
        <label>  
            <i class="fas fa-arrow-up"></i> Верх:  
            <input type="range" class="ползунок" id="paddingTop" min="0" max="100"  
value="40">  
            <span class="значение" id="paddingTopValue">40px</span>  
        </label>  
        <label>  
            <i class="fas fa-arrows-alt-h"></i> По бокам:  
            <input type="range" class="ползунок" id="paddingSides" min="0" max="100"  
value="30">  
            <span class="значение" id="paddingSidesValue">30px</span>  
        </label>  
    </div>  
  
    <div class="регулятор-группа">  
        <h4>MARGIN (внешние отступы)</h4>  
        <label>  
            <i class="fas fa-expand-alt"></i> Вокруг:  
            <input type="range" class="ползунок" id="marginAll" min="0" max="100"  
value="20">  
            <span class="значение" id="marginAllValue">20px</span>  
        </label>  
        <label>  
            <i class="fas fa-arrows-alt-v"></i> Между элементами:  
        </label>  
    </div>  
</div>
```

```
<input type="range" class="ползунок" id="marginBetween" min="5" max="50"
value="15">
    <span class="значение" id="marginBetweenValue">15px</span>
</label>
</div>

<div class="регулятор-группа">
    <h4>Быстрые пресеты</h4>
    <button class="кнопка-пресет" data-
пресет="компактный">Компактный</button>
        <button class="кнопка-пресет" data-пресет="воздушный">Воздушный</button>
        <button class="кнопка-пресет" data-пресет="роскошный">Роскошный</button>
        <button class="кнопка-пресет" onclick="сброситьОтступы()">Сбросить</button>
    </div>
</div>

<div class="подсказка-панели">
    <p><i class="fas fa-lightbulb"></i> <strong>Подсказка:</strong>
    ПADDING — воздух внутри визитки, MARGIN — расстояние от визитки до края
страницы.</p>
    </div>
</div>

<!-- Подсказки и инструкции --&gt;
&lt;div class="инструкция"&gt;
    &lt;h3&gt;&lt;i class="fas fa-graduation-cap"&gt;&lt;/i&gt; Что вы научитесь делать:&lt;/h3&gt;
    &lt;div class="навыки"&gt;
        &lt;div class="навык"&gt;
            &lt;i class="fas fa-arrows-alt"&gt;&lt;/i&gt;
            &lt;h4&gt;Управлять отступами&lt;/h4&gt;
            &lt;p&gt;Поймёте разницу между margin и padding&lt;/p&gt;
        &lt;/div&gt;
        &lt;div class="навык"&gt;
            &lt;i class="fas fa-palette"&gt;&lt;/i&gt;
            &lt;h4&gt;Подбирать цвета&lt;/h4&gt;
            &lt;p&gt;Научитесь создавать гармоничные цветовые схемы&lt;/p&gt;
        &lt;/div&gt;
        &lt;div class="навык"&gt;
            &lt;i class="fas fa-font"&gt;&lt;/i&gt;
            &lt;h4&gt;Работать со шрифтами&lt;/h4&gt;
            &lt;p&gt;Узнаете, как подключать красивые шрифты из интернета&lt;/p&gt;
        &lt;/div&gt;
        &lt;div class="навык"&gt;
            &lt;i class="fas fa-print"&gt;&lt;/i&gt;
        &lt;/div&gt;
    &lt;/div&gt;
&lt;/div&gt;</pre>
```

```

<h4>Создавать для печати</h4>
<p>Сделаете визитку, которую можно распечатать</p>
</div>
</div>

<div class="действия">
    <button class="кнопка-действия" onclick="показатьКод()">
        <i class="fas fa-code"></i> Показать HTML/CSS код
    </button>
    <button class="кнопка-действия" onclick="распечатать()">
        <i class="fas fa-print"></i> Распечатать визитку
    </button>
    <button class="кнопка-действия" onclick="сохранитьКакКартинку()">
        <i class="fas fa-download"></i> Сохранить как картинку
    </button>
</div>
</div>

<script src="script.js"></script>
</body>
</html>

```

---

## ШАГ 4: Создаём стили для ОБОИХ проектов

**Файл** style.css:

```

css

/* =====
   ОСНОВНЫЕ НАСТРОЙКИ ДЛЯ ВСЕХ ПРОЕКТОВ
===== */

/* ВАЖНОЕ ПРАВИЛО №1: box-sizing */
* {
    box-sizing: border-box;
    margin: 0;
    padding: 0;
}

/* Настойки всей страницы */
body {
    font-family: 'Open Sans', sans-serif;
}

```

```
line-height: 1.6;
color: #333;
background: linear-gradient(135deg, #f5f7fa 0%, #e4efe9 100%);
min-height: 100vh;
padding: 20px;
display: flex;
flex-direction: column;
align-items: center;
justify-content: flex-start;
}
```

```
/* Заголовок выбора проекта */
.выбор-проекта {
    background: white;
    padding: 30px;
    border-radius: 20px;
    box-shadow: 0 10px 30px rgba(0,0,0,0.1);
    text-align: center;
    margin-bottom: 30px;
    max-width: 800px;
    width: 100%;
    border: 3px solid #3498db;
}
```

```
.выбор-проекта h1 {
    font-family: 'Cormorant Garamond', serif;
    color: #2c3e50;
    margin-bottom: 15px;
    font-size: 2.5rem;
}
```

```
.кнопки-выбора {
    display: flex;
    gap: 20px;
    justify-content: center;
    margin-top: 25px;
    flex-wrap: wrap;
}
```

```
.кнопка-проекта {
    background: #f8f9fa;
    border: 2px solid #3498db;
    color: #3498db;
    padding: 15px 30px;
```

```
border-radius: 50px;
font-size: 18px;
cursor: pointer;
transition: all 0.3s;
display: flex;
align-items: center;
gap: 10px;
font-weight: 600;
}

.кнопка-проекта:hover {
background: #3498db;
color: white;
transform: translateY(-3px);
}

.кнопка-проекта.активная {
background: #3498db;
color: white;
box-shadow: 0 5px 15px rgba(52, 152, 219, 0.3);
}

/* Основной контейнер для проектов */
.контейнер {
max-width: 800px;
width: 100%;
margin-bottom: 40px;
transition: all 0.3s;
}

/*
===== СТИЛИ ДЛЯ ВИЗИТНОЙ КАРТОЧКИ =====
===== */

.визитка {
background: white;
border-radius: 20px;
box-shadow: 0 15px 40px rgba(0,0,0,0.15);
overflow: hidden;
transition: all 0.3s;
position: relative;
border: 1px solid rgba(0,0,0,0.1);
}

/*
Декоративный уголок визитки */

```

```
.визитка::before {
    content: "";
    position: absolute;
    top: 0;
    right: 0;
    width: 80px;
    height: 80px;
    background: linear-gradient(135deg, transparent 50%, #f8f9fa 50%);
    z-index: 1;
}

/* Шапка визитки */
.шапка-визитки {
    background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);
    color: white;
    text-align: center;
    padding: 40px 30px 30px 30px; /* PADDING пример! */
    position: relative;
    overflow: hidden;
}

.шапка-визитки::before {
    content: "";
    position: absolute;
    top: -50px;
    right: -50px;
    width: 150px;
    height: 150px;
    background: rgba(255,255,255,0.1);
    border-radius: 50%;
}

.аватарка {
    width: 120px;
    height: 120px;
    border-radius: 50%;
    border: 5px solid white;
    margin: 0 auto 20px auto; /* MARGIN пример! */
    background: #f8f9fa;
    display: flex;
    align-items: center;
    justify-content: center;
    font-size: 50px;
    color: #764ba2;
```

```
    box-shadow: 0 5px 15px rgba(0,0,0,0.2);
    position: relative;
    z-index: 2;
}

.имя-визитка {
    font-family: 'Cormorant Garamond', serif;
    font-size: 2.5rem;
    font-weight: 600;
    margin-bottom: 10px; /* MARGIN пример! */
}

.профессия {
    font-size: 1.2rem;
    opacity: 0.9;
    font-weight: 300;
}

/* Тело визитки */
.тело-визитки {
    padding: 30px; /* PADDING пример! */
    display: grid;
    grid-template-columns: 1fr 1fr;
    gap: 30px; /* Это тоже отступы между колонками! */
}

.колонка {
    display: flex;
    flex-direction: column;
    gap: 20px; /* Отступы между элементами внутри колонки */
}

.секция {
    background: #f8f9fa;
    padding: 20px; /* PADDING пример! */
    border-radius: 10px;
    border-left: 5px solid #3498db;
}

.секция h3 {
    color: #2c3e50;
    margin-bottom: 15px; /* MARGIN пример! */
    display: flex;
    align-items: center;
```

```
gap: 10px;
font-size: 1.2rem;
}

.секция h3 i {
    color: #3498db;
}

.контакт, .навык-элемент, .хобби-элемент {
    display: flex;
    align-items: center;
    gap: 15px;
    margin-bottom: 12px; /* MARGIN пример! */
    padding: 8px 0; /* PADDING пример! */
}

.контакт:last-child, .навык-элемент:last-child {
    margin-bottom: 0;
}

.иконка-контакта {
    width: 40px;
    height: 40px;
    background: white;
    border-radius: 50%;
    display: flex;
    align-items: center;
    justify-content: center;
    color: #3498db;
    font-size: 18px;
    box-shadow: 0 3px 10px rgba(0,0,0,0.1);
}

.текст-контакта {
    flex-grow: 1;
}

.метка {
    font-size: 0.9rem;
    color: #666;
    margin-bottom: 3px;
}

.значение {
```

```
font-weight: 600;
color: #2c3e50;
}

/* Прогресс-бар для навыков */
.прогресс-бар {
background: #e9ecf;
height: 10px;
border-radius: 5px;
overflow: hidden;
margin-top: 8px;
}

.прогресс-заполнение {
height: 100%;
background: linear-gradient(90deg, #3498db, #2ecc71);
border-radius: 5px;
}

/* Подвал визитки */
.подвал-визитки {
background: #2c3e50;
color: white;
padding: 20px 30px; /* PADDING пример! */
text-align: center;
border-top: 3px solid #3498db;
}

.подпись-визитка {
font-style: italic;
opacity: 0.8;
font-size: 0.95rem;
}

/*
=====
СТИЛИ ДЛЯ ОТКРЫТКИ
=====
*/
.открытка {
background: white;
border-radius: 20px;
box-shadow: 0 15px 40px rgba(0,0,0,0.15);
overflow: hidden;
transition: all 0.3s;
position: relative;
```

```
border: 1px solid rgba(0,0,0,0.1);
min-height: 600px;
}

/* Декоративные элементы открытки */
.узор-уголок {
    position: absolute;
    width: 100px;
    height: 100px;
    opacity: 0.1;
}

.узор-1 {
    top: 20px;
    left: 20px;
    border-top: 3px solid #e74c3c;
    border-left: 3px solid #e74c3c;
}

.узор-2 {
    bottom: 20px;
    right: 20px;
    border-bottom: 3px solid #3498db;
    border-right: 3px solid #3498db;
}

/* Шапка открытки */
.шапка-открытки {
    background: linear-gradient(135deg, #ff9a9e 0%, #fad0c4 100%);
    color: #d35400;
    text-align: center;
    padding: 40px 30px 30px 30px;
    position: relative;
    overflow: hidden;
}

.заголовок-открытка {
    font-family: 'Marck Script', cursive;
    font-size: 3rem;
    margin-bottom: 10px;
    text-shadow: 2px 2px 4px rgba(0,0,0,0.1);
}

.подзаголовок-открытка {
```

```
font-size: 1.3rem;
opacity: 0.9;
}

/* Тело открытки */
.тело-открытки {
    padding: 40px 30px;
    display: flex;
    flex-direction: column;
    gap: 25px;
}

.приветствие-открытка {
    font-family: 'Cormorant Garamond', serif;
    font-size: 1.8rem;
    color: #8b4513;
    line-height: 1.8;
    text-align: center;
    padding: 25px;
    background: #ffffaf0;
    border-radius: 15px;
    border-left: 5px solid #d4a762;
    position: relative;
}

.приветствие-открытка::before {
    content: "“";
    position: absolute;
    top: 10px;
    left: 15px;
    font-size: 3rem;
    color: #d4a762;
    opacity: 0.3;
}

.основной-текст-открытка {
    font-size: 1.2rem;
    line-height: 1.7;
    color: #5d4037;
    text-align: justify;
    hyphens: auto;
}

.абзац-открытка {
```

```
margin-bottom: 20px;
text-indent: 30px;
}

.абзац-открытка:last-child {
margin-bottom: 0;
}

.выделение-открытка {
font-weight: 600;
color: #e74c3c;
background: linear-gradient(120deg, #ffebee 0%, #ffebee 100%);
background-repeat: no-repeat;
background-size: 100% 0.4em;
background-position: 0 88%;
}

/* Пожелания в открытке */

.пожелания-открытка {
background: linear-gradient(135deg, #a1c4fd 0%, #c2e9fb 100%);
padding: 25px;
border-radius: 15px;
margin: 20px 0;
}

.пожелания-открытка h3 {
color: #2c3e50;
margin-bottom: 20px;
text-align: center;
display: flex;
align-items: center;
justify-content: center;
gap: 15px;
}

.список-пожеланий {
display: grid;
grid-template-columns: repeat(auto-fit, minmax(200px, 1fr));
gap: 15px;
}

.пожелание-элемент {
background: white;
padding: 15px;
```

```
border-radius: 10px;  
display: flex;  
align-items: center;  
gap: 15px;  
box-shadow: 0 3px 10px rgba(0,0,0,0.05);  
}
```

```
.пожелание-элемент i {  
color: #3498db;  
font-size: 20px;  
}
```

```
/* Подпись в открытке */  
.подпись-открытка {  
text-align: right;  
margin-top: 30px;  
padding-top: 20px;  
border-top: 2px dashed #ddd;  
}
```

```
.текст-подписи {  
font-family: 'Marck Script', cursive;  
font-size: 2rem;  
color: #8b4513;  
line-height: 1.4;  
}
```

```
.имя-открытка {  
font-size: 2.2rem;  
font-weight: bold;  
display: block;  
margin-top: 10px;  
}
```

```
/* Подвал открытки */  
.подвал-открытки {  
background: #34495e;  
color: white;  
padding: 20px 30px;  
text-align: center;  
border-top: 3px solid #3498db;  
font-size: 0.9rem;  
opacity: 0.8;  
}
```

```
/* =====
ПАНЕЛЬ УПРАВЛЕНИЯ
===== */

.панель-управления {
    background: white;
    padding: 25px;
    border-radius: 15px;
    box-shadow: 0 10px 30px rgba(0,0,0,0.1);
    max-width: 800px;
    width: 100%;
    margin-bottom: 30px;
    border: 2px solid #3498db;
}

.панель-управления h3 {
    color: #2c3e50;
    margin-bottom: 15px;
    display: flex;
    align-items: center;
    gap: 10px;
}

.регуляторы {
    display: grid;
    grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));
    gap: 25px;
    margin: 25px 0;
}

.регулятор-группа {
    background: #f8f9fa;
    padding: 20px;
    border-radius: 10px;
}

.регулятор-группа h4 {
    color: #2c3e50;
    margin-bottom: 15px;
    padding-bottom: 10px;
    border-bottom: 2px solid #e9ecf;
}

.регулятор-группа label {
```

```
display: block;
margin: 15px 0;
color: #495057;
display: flex;
align-items: center;
gap: 10px;
}

.ползунок {
  flex-grow: 1;
  height: 8px;
  border-radius: 4px;
  background: #dee2e6;
  outline: none;
  -webkit-appearance: none;
}

.ползунок::-webkit-slider-thumb {
  -webkit-appearance: none;
  width: 20px;
  height: 20px;
  border-radius: 50%;
  background: #3498db;
  cursor: pointer;
  border: 2px solid white;
  box-shadow: 0 2px 5px rgba(0,0,0,0.2);
}

.значение {
  font-family: 'Courier New', monospace;
  background: #2c3e50;
  color: white;
  padding: 3px 8px;
  border-radius: 4px;
  min-width: 50px;
  text-align: center;
  font-weight: bold;
}

.кнопка-пресет {
  background: #6c757d;
  color: white;
  border: none;
  padding: 10px 15px;
}
```

```
margin: 5px;
border-radius: 6px;
cursor: pointer;
font-size: 14px;
transition: all 0.3s;

}

.кнопка-пресет:hover {
    background: #545b62;
    transform: translateY(-2px);
}

.подсказка-панели {
    background: #e8f4fc;
    padding: 15px;
    border-radius: 8px;
    margin-top: 20px;
    border-left: 4px solid #3498db;
    font-size: 0.95rem;
}

.подсказка-панели i {
    color: #3498db;
    margin-right: 10px;
}

/*
=====
ИНСТРУКЦИЯ И КНОПКИ ДЕЙСТВИЙ
=====
*/
.инструкция {
    background: white;
    padding: 30px;
    border-radius: 15px;
    box-shadow: 0 10px 30px rgba(0,0,0,0.1);
    max-width: 800px;
    width: 100%;
}

.инструкция h3 {
    color: #2c3e50;
    margin-bottom: 25px;
    display: flex;
    align-items: center;
    gap: 10px;
}
```

```
}

.навыки {
    display: grid;
    grid-template-columns: repeat(auto-fit, minmax(180px, 1fr));
    gap: 20px;
    margin: 25px 0;
}

.навык {
    background: #f8f9fa;
    padding: 20px;
    border-radius: 10px;
    text-align: center;
    transition: transform 0.3s;
}

.навык:hover {
    transform: translateY(-5px);
    box-shadow: 0 5px 15px rgba(0,0,0,0.1);
}

.навык i {
    font-size: 2rem;
    color: #3498db;
    margin-bottom: 15px;
}

.навык h4 {
    color: #2c3e50;
    margin-bottom: 10px;
}

.навык p {
    color: #666;
    font-size: 0.9rem;
    line-height: 1.5;
}

.действия {
    display: flex;
    flex-wrap: wrap;
    gap: 15px;
    justify-content: center;
```

```
margin-top: 30px;
}

.кнопка-действия {
    background: linear-gradient(135deg, #3498db 0%, #2980b9 100%);
    color: white;
    border: none;
    padding: 15px 25px;
    border-radius: 50px;
    cursor: pointer;
    font-size: 16px;
    font-weight: 600;
    display: flex;
    align-items: center;
    gap: 10px;
    transition: all 0.3s;
    box-shadow: 0 5px 15px rgba(52, 152, 219, 0.3);
}

.кнопка-действия:hover {
    transform: translateY(-3px);
    box-shadow: 0 8px 20px rgba(52, 152, 219, 0.4);
}

.кнопка-действия:active {
    transform: translateY(-1px);
}

/* =====
АДАПТИВНОСТЬ ДЛЯ МОБИЛЬНЫХ
===== */
@media (max-width: 768px) {
    body {
        padding: 10px;
    }

    .выбор-проекта, .панель-управления, .инструкция {
        padding: 20px;
    }

    .кнопки-выбора {
        flex-direction: column;
    }
}
```

```
.кнопка-проекта {
    width: 100%;
    justify-content: center;
}

.тело-визитки {
    grid-template-columns: 1fr;
    gap: 20px;
}

.имя-визитка {
    font-size: 2rem;
}

.заголовок-открытка {
    font-size: 2.5rem;
}

.приветствие-открытка {
    font-size: 1.5rem;
    padding: 20px;
}

.регуляторы {
    grid-template-columns: 1fr;
}

.навыки {
    grid-template-columns: repeat(2, 1fr);
}

.действия {
    flex-direction: column;
}

.кнопка-действия {
    width: 100%;
    justify-content: center;
}

}

@media (max-width: 480px) {
    .навыки {
        grid-template-columns: 1fr;
```

```
}

.имя-визитка {
    font-size: 1.8rem;
}

.заголовок-открытка {
    font-size: 2rem;
}

}
```

---

## ШАГ 5: Создаём JavaScript для интерактивности

**Файл script.js:**

```
javascript

// =====
// ОСНОВНЫЕ ПЕРЕМЕННЫЕ И КОНСТАНТЫ
// =====

// HTML-код для визитной карточки
const ВИЗИТКА_HTML = `

<div class="визитка" id="визитка">

    <!-- Шапка визитки -->
    <div class="шапка-визитки">
        <div class="аватарка">
            <i class="fas fa-user"></i>
        </div>
        <h1 class="имя-визитка">Галина Ивановна</h1>
        <p class="профессия">Садовод-любитель | Рукодельница | Бабушка-эксперт</p>
    </div>

    <!-- Тело визитки -->
    <div class="тело-визитки">
        <!-- Левая колонка -->
        <div class="колонка">
            <div class="секция">
                <h3><i class="fas fa-address-card"></i> Контакты</h3>
                <div class="контакт">
                    <div class="иконка-контакта">
                        <i class="fas fa-phone"></i>
                    </div>
                </div>
            </div>
        </div>
    </div>
</div>
```

```
</div>
<div class="текст-контакта">
    <div class="метка">Телефон</div>
    <div class="значение">+7 (999) 123-45-67</div>
</div>
</div>
<div class="контакт">
    <div class="иконка-контакта">
        <i class="fas fa-envelope"></i>
    </div>
    <div class="текст-контакта">
        <div class="метка">Email</div>
        <div class="значение">galina.sadovod@example.ru</div>
    </div>
</div>
<div class="контакт">
    <div class="иконка-контакта">
        <i class="fas fa-map-marker-alt"></i>
    </div>
    <div class="текст-контакта">
        <div class="метка">Адрес</div>
        <div class="значение">г. Москва, ул. Садовая, 15</div>
    </div>
</div>
</div>

<div class="секция">
    <h3><i class="fas fa-heart"></i> Хобби и увлечения</h3>
    <div class="хобби-элемент">
        <i class="fas fa-seedling" style="color: #27ae60;"></i>
        <span>Садоводство (40 лет опыта)</span>
    </div>
    <div class="хобби-элемент">
        <i class="fas fa-utensils" style="color: #e74c3c;"></i>
        <span>Кулинария (фирменные пироги)</span>
    </div>
    <div class="хобби-элемент">
        <i class="fas fa-knitting" style="color: #9b59b6;"></i>
        <span>Вязание (шарфы, носки, пледы)</span>
    </div>
    <div class="хобби-элемент">
        <i class="fas fa-book" style="color: #3498db;"></i>
        <span>Чтение (исторические романы)</span>
    </div>

```

```
</div>
</div>

<!-- Правая колонка --&gt;
&lt;div class="колонка"&gt;
    &lt;div class="секция"&gt;
        &lt;h3&gt;&lt;i class="fas fa-star"&gt;&lt;/i&gt; Навыки и умения&lt;/h3&gt;
        &lt;div class="навык-элемент"&gt;
            &lt;span&gt;Выращивание роз&lt;/span&gt;
            &lt;div class="прогресс-бар"&gt;
                &lt;div class="прогресс-заполнение" style="width: 95%"&gt;&lt;/div&gt;
            &lt;/div&gt;
        &lt;/div&gt;
        &lt;div class="навык-элемент"&gt;
            &lt;span&gt;Консервирование&lt;/span&gt;
            &lt;div class="прогресс-бар"&gt;
                &lt;div class="прогресс-заполнение" style="width: 90%"&gt;&lt;/div&gt;
            &lt;/div&gt;
        &lt;/div&gt;
        &lt;div class="навык-элемент"&gt;
            &lt;span&gt;Вязание спицами&lt;/span&gt;
            &lt;div class="прогресс-бар"&gt;
                &lt;div class="прогресс-заполнение" style="width: 85%"&gt;&lt;/div&gt;
            &lt;/div&gt;
        &lt;/div&gt;
        &lt;div class="навык-элемент"&gt;
            &lt;span&gt;Создание веб-страниц&lt;/span&gt;
            &lt;div class="прогресс-бар"&gt;
                &lt;div class="прогресс-заполнение" style="width: 70%"&gt;&lt;/div&gt;
            &lt;/div&gt;
        &lt;/div&gt;
    &lt;/div&gt;
&lt;/div&gt;

&lt;div class="секция"&gt;
    &lt;h3&gt;&lt;i class="fas fa-comments"&gt;&lt;/i&gt; Обо мне&lt;/h3&gt;
    &lt;p style="line-height: 1.6; color: #555;"&gt;
        Привет! Я Галина — пенсионерка с активной жизненной позицией.
        Обожаю свой сад, где провожу большую часть времени с весны до осени.
        Люблю делиться знаниями о растениях и рецептами домашних заготовок.
        Недавно увлеклась веб-дизайном — оказывается, это очень увлекательно!
    &lt;/p&gt;
    &lt;div style="margin-top: 15px; padding: 10px; background: #fff8e1; border-radius: 5px; border-left: 4px solid #f39c12;"&gt;
        &lt;strong&gt;Девиз:&lt;/strong&gt; «Учиться никогда не поздно!»
    &lt;/div&gt;
&lt;/div&gt;</pre>
```

```
</div>
</div>
</div>
</div>

<!-- Подвал визитки -->
<div class="подвал-визитки">
    <p class="подпись-визитка">
        «Сад, как и жизнь, требует ухода, терпения и любви. И тогда он отвечает красотой и
изобилием.»
    </p>
</div>
</div>
`;

// HTML-код для поздравительной открытки
const ОТКРЫТКА_HTML = `

<div class="открытка" id="открытка">
    <!-- Декоративные уголки -->
    <div class="узор-уголок узор-1"></div>
    <div class="узор-уголок узор-2"></div>

    <!-- Шапка открытки -->
    <div class="шапка-открытки">
        <h1 class="заголовок-открытка">С Днём Рождения!</h1>
        <p class="подзаголовок-открытка">Дорогой внук Андрюша</p>
    </div>

    <!-- Тело открытки -->
    <div class="тело-открытки">
        <div class="приветствие-открытка">
            Пишу тебе эту открытку в твой особенный день — день, когда ты появился на свет и
сделал нашу жизнь ярче!
        </div>

        <div class="основной-текст-открытка">
            <p class="абзац-открытка">
                Помнишь, как ты в пять лет приезжал ко мне на дачу и мы вместе сажали
яблоню?
            <span class="выделение-открытка">Ты тогда так серьёзно копал лунку, точно
знал, что это дерево будет расти вместе с тобой.</span>
                Сейчас эта яблоня уже большая, каждую осень даёт сладкие плоды, и я всегда
вспоминаю тебя, когда их собираю.
            </p>
        </div>
    </div>
</div>
```

```
<p class="абзац-открытка">
    Смотрю на тебя сейчас — взрослый, самостоятельный, умный молодой
    человек — и сердце наполняется гордостью.

    Ты многое добился, и я уверена, что впереди тебя ждут ещё большие успехи.

    Но помни, пожалуйста, самое главное: <span class="выделение-
открытка">счастье — не только в достижениях, но и в простых радостях жизни.</span>

</p>
</div>

<!-- Пожелания -->
<div class="пожелания-открытка">
    <h3><i class="fas fa-gift"></i> Мои пожелания тебе:</h3>
    <div class="список-пожеланий">
        <div class="пожелание-элемент">
            <i class="fas fa-heartbeat"></i>
            <div>
                <strong>Здоровья</strong><br>
                <small>Крепкого, как дуб</small>
            </div>
        </div>
        <div class="пожелание-элемент">
            <i class="fas fa-laugh-beam"></i>
            <div>
                <strong>Радости</strong><br>
                <small>Каждый день</small>
            </div>
        </div>
        <div class="пожелание-элемент">
            <i class="fas fa-briefcase"></i>
            <div>
                <strong>Успехов в работе</strong><br>
                <small>Но без переутомления</small>
            </div>
        </div>
        <div class="пожелание-элемент">
            <i class="fas fa-users"></i>
            <div>
                <strong>Верных друзей</strong><br>
                <small>И любящей семьи</small>
            </div>
        </div>
    </div>
</div>
```

```
<!-- Подпись -->
<div class="подпись-открытка">
    <div class="текст-подписи">
        Целую крепко!<br>
        <span class="имя-открытка">Твоя бабушка Галина</span>
    </div>
</div>
</div>

<!-- Подвал открытки -->
<div class="подвал-открытки">
    <p>P.S. Жду тебя летом — яблоки уже спевают!</p>
</div>
</div>
`;

// =====
// ПЕРЕМЕННЫЕ И СОСТОЯНИЕ
// =====

let текущийПроект = 'визитка'; // 'визитка' или 'открытка'
let текущийЭлемент = null; // Текущий элемент (визитка или открытка)

// =====
// ОСНОВНЫЕ ФУНКЦИИ
// =====

// Загрузка проекта при загрузке страницы
window.onload = function() {
    // Загружаем визитку по умолчанию
    загрузитьПроект('визитка');

    // Настраиваем обработчики для кнопок выбора проекта
    document.querySelectorAll('.кнопка-проекта').forEach(кнопка => {
        кнопка.addEventListener('click', function() {
            const проект = this.getAttribute('data-проект');
            загрузитьПроект(проект);
        });
    });

    // Настраиваем обработчики для ползунков
    настроитьПолзунки();
}
```

```
// Настраиваем обработчики для пресетов
настроитьПресеты();

// Загрузка выбранного проекта
function загрузитьПроект(проект) {
    текущийПроект = проект;
    const контейнер = document.getElementById('контейнерПроекта');

    // Очищаем контейнер
    контейнер.innerHTML = '';

    // Загружаем соответствующий HTML
    if (проект === 'визитка') {
        контейнер.innerHTML = ВИЗИТКА_HTML;
        текущийЭлемент = document.getElementById('визитка');
    } else {
        контейнер.innerHTML = ОТКРЫТКА_HTML;
        текущийЭлемент = document.getElementById('открытка');
    }

    // Обновляем активную кнопку
    document.querySelectorAll('.кнопка-проекта').forEach(кнопка => {
        кнопка.classList.remove('активная');
        if (кнопка.getAttribute('data-проект') === проект) {
            кнопка.classList.add('активная');
        }
    });
}

// Применяем текущие значения отступов
применитьОтступы();
}

// Настройка ползунков
function настроитьПолзунки() {
    // Padding top
    document.getElementById('paddingTop').addEventListener('input', function() {
        const значение = this.value + 'px';
        document.getElementById('paddingTopValue').textContent = значение;
        применитьОтступы();
    });
}

// Padding sides
document.getElementById('paddingSides').addEventListener('input', function() {
```

```

const значение = this.value + 'px';
document.getElementById('paddingSidesValue').textContent = значение;
применитьОтступы();
});

// Margin all
document.getElementById('marginAll').addEventListener('input', function() {
    const значение = this.value + 'px';
    document.getElementById('marginAllValue').textContent = значение;
    применитьОтступы();
});

// Margin between
document.getElementById('marginBetween').addEventListener('input', function() {
    const значение = this.value + 'px';
    document.getElementById('marginBetweenValue').textContent = значение;
    применитьОтступы();
});
}

// Настройка пресетов
function настроитьПресеты() {
    document.querySelectorAll('.кнопка-пресет').forEach(кнопка => {
        кнопка.addEventListener('click', function() {
            const пресет = this.getAttribute('data-пресет');

            switch(пресет) {
                case 'компактный':
                    установитьОтступы(20, 15, 10, 10);
                    break;
                case 'воздушный':
                    установитьОтступы(50, 40, 40, 30);
                    break;
                case 'роскошный':
                    установитьОтступы(80, 60, 60, 40);
                    break;
            }
        });
    });
}

// Установка значений отступов
function установитьОтступы(paddingTop, paddingSides, marginAll, marginBetween) {
    document.getElementById('paddingTop').value = paddingTop;

```

```
document.getElementById('paddingTopValue').textContent = paddingTop + 'px';

document.getElementById('paddingSides').value = paddingSides;
document.getElementById('paddingSidesValue').textContent = paddingSides + 'px';

document.getElementById('marginAll').value = marginAll;
document.getElementById('marginAllValue').textContent = marginAll + 'px';

document.getElementById('marginBetween').value = marginBetween;
document.getElementById('marginBetweenValue').textContent = marginBetween + 'px';

применитьОтступы();

}

// Применение отступов к текущему проекту
function применитьОтступы() {
    if (!текущийЭлемент) return;

    const paddingTop = document.getElementById('paddingTop').value + 'px';
    const paddingSides = document.getElementById('paddingSides').value + 'px';
    const marginAll = document.getElementById('marginAll').value + 'px';
    const marginBetween = document.getElementById('marginBetween').value + 'px';

    // Применяем margin к основному элементу
    текущийЭлемент.style.margin = marginAll;

    // Применяем padding к шапке
    const шапка = текущийЭлемент.querySelector('.шапка-визитки, .шапка-открытки');
    if (шапка) {
        шапка.style.paddingTop = paddingTop;
        шапка.style.paddingLeft = paddingSides;
        шапка.style.paddingRight = paddingSides;
    }

    // Применяем padding к телу
    const тело = текущийЭлемент.querySelector('.тело-визитки, .тело-открытки');
    if (тело) {
        тело.style.padding = paddingTop + ' ' + paddingSides;
    }

    // Применяем отступы между элементами
    if (текущийПроект === 'визитка') {
        // Для визитки
        const секции = текущийЭлемент.querySelectorAll('.секция');
```

```

секции.forEach(секция => {
    секция.style.marginBottom = marginBetween;
});

const контакты = текущийЭлемент.querySelectorAll('.контакт, .навык-элемент, .хобби-элемент');
контакты.forEach(элемент => {
    элемент.style.marginBottom = marginBetween;
});
} else {
    // Для открытки
    const абзацы = текущийЭлемент.querySelectorAll('.абзац-открытка');
    абзацы.forEach(абзац => {
        абац.style.marginBottom = marginBetween;
    });
}

const пожелания = текущийЭлемент.querySelectorAll('.пожелание-элемент');
пожелания.forEach(элемент => {
    элемент.style.marginBottom = marginBetween;
});
}

// Сброс отступов
function сброситьОтступы() {
    установитьОтступы(40, 30, 20, 15);
}

// Показать HTML/CSS код
function показатьКод() {
    if (!текущийЭлемент) return;

    const paddingTop = document.getElementById('paddingTop').value;
    const paddingSides = document.getElementById('paddingSides').value;
    const marginAll = document.getElementById('marginAll').value;
    const marginBetween = document.getElementById('marginBetween').value;

    const cssКод = `.мой-проект {
        /* Основные отступы */
        margin: ${marginAll}px; /* Внешние отступы вокруг всего элемента */
        padding: ${paddingTop}px ${paddingSides}px; /* Внутренние отступы */

        /* Отступы между внутренними элементами */
    }`;
}

```

```
.внутренний-элемент {
    margin-bottom: ${marginBetween}px; /* Отступ между элементами внутри */
}

const htmlКод = текущийЭлемент.outerHTML
    .replace(/</g, '&lt;')
    .replace(/>/g, '&gt;')
    .replace(/\n/g, '<br>')
    .replace(/\s/g, '&nbsp;&nbsp;');

const модалка = `
<div style="
    position: fixed;
    top: 0;
    left: 0;
    width: 100%;
    height: 100%;
    background: rgba(0,0,0,0.8);
    display: flex;
    align-items: center;
    justify-content: center;
    z-index: 1000;
    padding: 20px;
">
    <div style="
        background: white;
        padding: 30px;
        border-radius: 15px;
        max-width: 800px;
        width: 100%;
        max-height: 80vh;
        overflow-y: auto;
">
        <h3 style="color: #2c3e50; margin-top: 0;">📋 Код вашего проекта</h3>

        <div style="margin-bottom: 25px;">
            <h4>CSS-код для отступов:</h4>
            <pre style="
                background: #2c3e50;
                color: white;
                padding: 20px;
                border-radius: 8px;
                overflow-x: auto;
            ">
${htmlКод}
            </pre>
        </div>
    </div>
</div>
```

```
        font-family: 'Courier New', monospace;
    ">${cssКод}</pre>
</div>

<div>
    <h4>HTML-код ${текущийПроект === 'визитка' ? 'визитки' :
'открытки'}:</h4>
    <pre style="
        background: #f8f9fa;
        padding: 20px;
        border-radius: 8px;
        overflow-x: auto;
        border: 1px solid #ddd;
        font-family: 'Courier New', monospace;
        max-height: 300px;
        overflow-y: auto;
    ">${htmlКод}</pre>
</div>

<div style="margin-top: 25px; text-align: center;">
    <button onclick="this.parentElement.parentElement.parentElement.remove()" style="
        background: #3498db;
        color: white;
        border: none;
        padding: 12px 30px;
        border-radius: 6px;
        cursor: pointer;
        font-size: 16px;
    ">
        Закрыть
    </button>
</div>
</div>
</div>
';

document.body.insertAdjacentHTML('beforeend', модалка);
}

// Функция печати
function распечатать() {
    if (!текущийЭлемент) return;
```

```
// Создаём новое окно для печати
const окноПечати = window.open("", '_blank');

const стили = Array.from(document.styleSheets)
.map(styleSheet => {
    try {
        return Array.from(styleSheet.cssRules || [])
            .map(rule => rule.cssText)
            .join('\n');
    } catch {
        return "";
    }
})
.join('\n');

const htmlДляПечати = `
<!DOCTYPE html>
<html>
<head>
    <title>Моя ${текущийПроект === 'визитка' ? 'визитная карточка' : 'поздравительная
открытка'}</title>
    <style>
        ${стили}
        body {
            padding: 50px !important;
            background: white !important;
        }
        .выбор-проекта,
        .панель-управления,
        .инструкция {
            display: none !important;
        }
        .визитка, .открытка {
            box-shadow: none !important;
            border: 1px solid #ddd !important;
            margin: 0 !important;
        }
    @media print {
        @page {
            margin: 20mm;
        }
        body {
            padding: 0 !important;
        }
    }

```

```

        }
    </style>
</head>
<body>
${текущийЭлемент.outerHTML}
<script>
    window.onload = function() {
        window.print();
        setTimeout(() => window.close(), 1000);
    };
</script>
</body>
</html>
';

окноПечати.document.write(htmlДляПечати);
окноПечати.document.close();
}

// Функция сохранения как картинки (упрощённая версия)
function сохранитьКакКартинку() {
    alert('Для сохранения как картинки:\n\n1. Нажмите Print Screen (PrtScn) на клавиатуре\n\n2. Откройте Paint или другой редактор\n\n3. Нажмите Ctrl+V (вставить)\n\n4. Обрежьте и сохраните\n\nИли используйте браузерные расширения для скриншотов.');
}

// =====
// ДОПОЛНИТЕЛЬНЫЕ ВОЗМОЖНОСТИ ДЛЯ ЭКСПЕРИМЕНТОВ
// =====

// Функция для изменения цвета темы
function изменитьЦветТемы(цвет) {
    if (!текущийЭлемент) return;

    const основныеЦвета = {
        'синий': { основной: '#3498db', градиент: 'linear-gradient(135deg, #667eea 0%, #764ba2 100%)' },
        'зелёный': { основной: '#27ae60', градиент: 'linear-gradient(135deg, #56ab2f 0%, #a8e063 100%)' },
        'оранжевый': { основной: '#e67e22', градиент: 'linear-gradient(135deg, #f46b45 0%, #eea849 100%)' },
        'фиолетовый': { основной: '#9b59b6', градиент: 'linear-gradient(135deg, #7579ff 0%, #b224ef 100%)' }
    };
}

```

```

const цвета = основныеЦвета[цвет] || основныеЦвета['синий'];

// Изменяем цвета в визитке
if (текущийПроект === 'визитка') {
    const шапка = текущийЭлемент.querySelector('.шапка-визитки');
    if (шапка) шапка.style.background = цвета.градиент;

    const границы = текущийЭлемент.querySelectorAll('.секция');
    границы.forEach(секция => {
        секция.style.borderLeftColor = цвета.основной;
    });

    const иконки = текущийЭлемент.querySelectorAll('.иконка-контакта');
    иконки.forEach(иконка => {
        иконка.style.color = цвета.основной;
    });

    const прогресс = текущийЭлемент.querySelectorAll('.прогресс-заполнение');
    прогресс.forEach(бар => {
        бар.style.background = `linear-gradient(90deg, ${цвета.основной}, #2ecc71}`;
    });
}

// Функция для изменения шрифта
function изменитьШрифт(шрифт) {
    if (!текущийЭлемент) return;

    const шрифты = {
        'классический': "Cormorant Garamond, serif",
        'рукописный': "'Marck Script', cursive",
        'современный': "'Open Sans', sans-serif",
        'строгий': "'Times New Roman', serif"
    };

    const выбранныйШрифт = шрифты[шрифт] || шрифты['классический'];

    if (текущийПроект === 'визитка') {
        const имя = текущийЭлемент.querySelector('.имя-визитка');
        if (имя) имя.style.fontFamily = выбранныйШрифт;
    } else {
        const заголовок = текущийЭлемент.querySelector('.заголовок-открытка');
        if (заголовок) заголовок.style.fontFamily = выбранныйШрифт;
    }
}

```

```
 }  
}
```

---

## ШАГ 6: Запускаем и наслаждаемся результатом!

### Что нужно сделать:

1. Сохраните все три файла (index.html, style.css, script.js) в одну папку
2. Откройте файл index.html в браузере
3. Вы увидите интерактивный интерфейс!

### Что можно делать:

#### A. С визитной карточкой:

##### 1. Экспериментируйте с отступами:

- Увеличьте PADDING — визитка станет "воздушнее"
- Увеличьте MARGIN — визитка отодвинется от краёв
- Уменьшите отступы — получите компактный вариант

##### 2. Используйте пресеты:

- **Компактный** — для экономии места
- **Воздушный** — для элегантного вида
- **Роскошный** — для особых случаев

##### 3. Персонализируйте:

- Измените имя и профессию в коде
- Добавьте реальные контакты
- Измените навыки и хобби под себя

#### B. С поздравительной открыткой:

1. **Настройте отступы** для идеального баланса текста
2. **Измените текст** поздравительного письма
3. **Подстройте пожелания** под конкретного человека
4. **Распечатайте** и подарите в реальной жизни!

## **Дополнительные возможности:**

1. **Показать код** — увидите HTML и CSS своего проекта
  2. **Распечатать** — получите версию для печати
  3. **Сохранить как картинку** — сделайте скриншот
- 

## **Чему вы научились в этом практикуме:**

- Понимание margin и padding на практике**
  - Создание адаптивных макетов** с помощью CSS Grid и Flexbox
  - Работа с градиентами** и цветовыми схемами
  - Подключение и использование** Google Fonts
  - Создание интерактивных элементов** с JavaScript
  - Верстка реальных проектов** — визиток и открыток
  - Подготовка к печати** — создание print-версий
  - Организация кода** в отдельные файлы (HTML, CSS, JS)
- 

## **Домашнее задание:**

### **Задание 1: Создайте свою личную визитку**

1. Измените все данные на свои
2. Добавьте реальную фотографию (замените иконку пользователя)
3. Подберите цветовую гамму, которая вам нравится
4. Добавьте раздел "Мои достижения" или "Мои проекты"

### **Задание 2: Создайте открытку для конкретного человека**

1. Напишите персональное письмо
2. Подберите цветовую гамму, подходящую получателю
3. Добавьте общую фотографию (если есть)
4. Создайте специальные пожелания только для этого человека

## **Задание 3: Усовершенствуйте код**

1. Добавьте кнопку "Изменить цвет темы"
2. Создайте возможность менять шрифты
3. Добавьте анимацию при наведении на элементы
4. Сделайте версию на английском языке

## **Задание 4: Создайте коллекцию**

1. Сделайте несколько визиток для разных целей:
    - Для садового клуба
    - Для вязального кружка
    - Для встречи выпускников
  2. Создайте галерею с переключением между ними
- 

## **Советы для успеха:**

1. **Не бойтесь экспериментировать** — меняйте значения и смотрите, что получается
2. **Сохраняйте разные версии** — создавайте копии файлов перед большими изменениями
3. **Используйте инструменты разработчика** (F12) для отладки
4. **Проверяйте на разных устройствах** — телефоне, планшете, компьютере
5. **Делитесь с друзьями** — получите обратную связь и вдохновение

**Поздравляю!** Вы только что создали свой первый полноценный веб-проект с интерактивными элементами! Это уже не просто учебный пример, а реальный инструмент, который можно использовать в жизни. В следующей главе мы научимся создавать красивые рамки и тени — это сделает ваши проекты ещё более профессиональными! 

## ● Глава 5: Рамки и тени – создаем настроение.

- Свойство **border**: сплошная, пунктирная, цвет, толщина. Делаем аккуратную рамку.

Дорогой друг, представьте, что вы выбираете рамку для любимой фотографии. Какую вы возьмёте? Тонкую золотую? Широкую деревянную? А может, разноцветную пунктирную для детского рисунка? В CSS свойство **border** — это именно тот инструмент, который позволяет создавать такие рамки для любых элементов на вашей веб-странице!

### 1. Что такое border и из чего он состоит?

**Border (граница, рамка)** — это свойство CSS, которое создаёт рамку ВОКРУГ элемента. Состоит из трёх основных характеристик:

text

```
border: 3px solid red;
```

↑      ↑      ↑

толщина    стиль    цвет

### Простейший пример:

css

```
.моя-рамка {  
    border: 2px solid black; /* Рамка: 2px толщиной, сплошная, чёрная */  
}
```

## 2. Три главные характеристики border

### A. Толщина (ширина) рамки

Определяет, насколько толстая будет рамка.

css

```
.тонкая-рамка {  
    border-width: 1px; /* Очень тонкая */  
}  
  
.стандартная-рамка {  
    border-width: 3px; /* Стандартная толщина */  
}  
  
.толстая-рамка {  
    border-width: 10px; /* Толстая, заметная */  
}  
  
.разная-толщина {  
    border-width: 5px 10px 15px 20px; /* top right bottom left */  
}
```

### B. Стиль рамки (самое интересное!)

Определяет, как выглядит линия рамки.

css

```
/* Сплошная линия (самая частая) */
```

```
.сплошная {  
    border-style: solid;  
}
```

```
/* Пунктирная (точки) */
```

```
.пунктирная {  
    border-style: dotted;  
}
```

```
/* Тире (короткие чёрточки) */
```

```
.тире {
```

```
border-style: dashed;  
}  
  
/* Двойная рамка */  
.двойная {  
    border-style: double;  
}  
  
/* Объёмная (вдавленная) */  
.объемная {  
    border-style: groove;  
}  
  
/* Рельефная (выпуклая) */  
.рельефная {  
    border-style: ridge;  
}  
  
/* Вставленная (как внутри) */  
.вставленная {  
    border-style: inset;  
}  
  
/* Выступающая (как наружу) */  
.выступающая {  
    border-style: outset;  
}
```

## С. Цвет рамки

Можно использовать любые цвета, которые мы уже изучили!

css

```
.черная {  
    border-color: black;  
}  
  
.красная {  
    border-color: #ff0000; /* HEX */  
}  
  
.синяя {
```

```
border-color: rgb(0, 0, 255); /* RGB */  
}  
  
.прозрачная {  
    border-color: rgba(0, 0, 0, 0.5); /* Полупрозрачная */  
}
```

### **3. Создаём интерактивную лабораторию рамок**

Создайте файл index.html:

html

```
<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Лаборатория рамок: Изучаем border</title>
    <style>
        /* ОСНОВНЫЕ НАСТРОЙКИ */
        * {
            box-sizing: border-box;
            margin: 0;
            padding: 0;
        }

        body {
            font-family: 'Arial', sans-serif;
            line-height: 1.6;
            color: #333;
            background: linear-gradient(135deg, #f5f7fa 0%, #c3cfe2 100%);
            min-height: 100vh;
            padding: 20px;
        }

        h1 {
            text-align: center;
            color: #2c3e50;
            margin: 30px 0;
            text-shadow: 1px 1px 2px rgba(0,0,0,0.1);
            font-size: 2.5rem;
        }
    

```

```
}

/* КОНТЕЙНЕРЫ */

.раздел {
    background: white;
    padding: 30px;
    border-radius: 20px;
    box-shadow: 0 10px 30px rgba(0,0,0,0.1);
    margin-bottom: 40px;
}

.раздел h2 {
    color: #2c3e50;
    margin-bottom: 25px;
    padding-bottom: 15px;
    border-bottom: 3px solid;
}

.инфо-блок {
    background: #e8f4fc;
    padding: 20px;
    border-radius: 10px;
    margin: 20px 0;
    border-left: 5px solid #3498db;
}

/* ГАЛЕРЕЯ РАМОК */

.галерея-рамок {
    display: grid;
    grid-template-columns: repeat(auto-fill, minmax(300px, 1fr));
    gap: 25px;
    margin-top: 30px;
}

.рамка-пример {
    background: white;
    padding: 30px;
    border-radius: 10px;
    text-align: center;
    transition: transform 0.3s, box-shadow 0.3s;
    cursor: pointer;
    border: 1px solid #e9ecef;
}
```

```
.рамка-пример:hover {
    transform: translateY(-5px);
    box-shadow: 0 15px 30px rgba(0,0,0,0.15);
}

.название-рамки {
    font-weight: bold;
    margin-bottom: 15px;
    color: #2c3e50;
    font-size: 1.2rem;
}

.код-рамки {
    background: #2c3e50;
    color: white;
    padding: 10px;
    border-radius: 5px;
    font-family: 'Courier New', monospace;
    font-size: 14px;
    margin-top: 15px;
    word-break: break-all;
}

/* ИНТЕРАКТИВНАЯ РАМКА */

.интерактивная-зона {
    display: grid;
    grid-template-columns: 1fr 1fr;
    gap: 40px;
    margin-top: 30px;
}

@media (max-width: 768px) {
    .интерактивная-зона {
        grid-template-columns: 1fr;
    }
}

.предпросмотр {
    background: #f8f9fa;
    border-radius: 15px;
    padding: 40px;
    display: flex;
    align-items: center;
    justify-content: center;
```

```
        min-height: 300px;  
        border: 2px dashed #dee2e6;  
    }
```

```
.образец-рамки {  
    width: 250px;  
    height: 200px;  
    background: white;  
    display: flex;  
    flex-direction: column;  
    align-items: center;  
    justify-content: center;  
    text-align: center;  
    padding: 20px;  
    transition: all 0.3s;  
}
```

```
.образец-рамки h3 {  
    margin-bottom: 15px;  
    color: #2c3e50;  
}
```

```
/* ПАНЕЛЬ УПРАВЛЕНИЯ */  
.панель-управления {  
    background: #f8f9fa;  
    padding: 30px;  
    border-radius: 15px;  
    border: 2px solid #e9ecf;  
}
```

```
.регуляторы {  
    display: flex;  
    flex-direction: column;  
    gap: 25px;  
}
```

```
.регулятор-группа {  
    background: white;  
    padding: 20px;  
    border-radius: 10px;  
    border: 1px solid #dee2e6;  
}
```

```
.регулятор-группа h3 {
```

```
        color: #2c3e50;
        margin-bottom: 15px;
        display: flex;
        align-items: center;
        gap: 10px;
    }

    .регулятор-группа h3 i {
        color: #3498db;
    }

    .стили-рамок {
        display: grid;
        grid-template-columns: repeat(3, 1fr);
        gap: 10px;
        margin-top: 15px;
    }

    .кнопка-стиль {
        padding: 12px;
        border: 2px solid #e9ecf;
        background: white;
        border-radius: 8px;
        cursor: pointer;
        font-size: 14px;
        transition: all 0.3s;
        text-align: center;
    }

    .кнопка-стиль:hover {
        border-color: #3498db;
        background: #f0f8ff;
    }

    .кнопка-стиль.active {
        border-color: #3498db;
        background: #3498db;
        color: white;
        font-weight: bold;
    }

    .цвета-рамок {
        display: grid;
        grid-template-columns: repeat(5, 1fr);
```

```
gap: 10px;
margin-top: 15px;
}

.knopka-цвет {
    width: 40px;
    height: 40px;
    border-radius: 50%;
    cursor: pointer;
    border: 3px solid white;
    box-shadow: 0 3px 10px rgba(0,0,0,0.1);
    transition: transform 0.3s;
}

.knopka-цвет:hover {
    transform: scale(1.1);
}

.knopka-цвет.active {
    border-color: #2c3e50;
    transform: scale(1.1);
}

.polzunok {
    width: 100%;
    margin: 15px 0;
    height: 8px;
    border-radius: 4px;
    background: #dee2e6;
    outline: none;
    -webkit-appearance: none;
}

.polzunok::-webkit-slider-thumb {
    -webkit-appearance: none;
    width: 20px;
    height: 20px;
    border-radius: 50%;
    background: #3498db;
    cursor: pointer;
    border: 3px solid white;
    box-shadow: 0 2px 5px rgba(0,0,0,0.2);
}
```

```
.значение {
    font-family: 'Courier New', monospace;
    background: #2c3e50;
    color: white;
    padding: 3px 10px;
    border-radius: 4px;
    display: inline-block;
    margin-left: 10px;
    font-weight: bold;
}

.кнопки-действий {
    display: flex;
    gap: 15px;
    margin-top: 25px;
    flex-wrap: wrap;
}

.кнопка-действия {
    background: #3498db;
    color: white;
    border: none;
    padding: 12px 24px;
    border-radius: 8px;
    cursor: pointer;
    font-size: 16px;
    display: flex;
    align-items: center;
    gap: 10px;
    transition: all 0.3s;
}

.кнопка-действия:hover {
    background: #2980b9;
    transform: translateY(-2px);
}

/* ПРАКТИЧЕСКИЕ ПРИМЕРЫ */
.примеры-практические {
    display: grid;
    grid-template-columns: repeat(auto-fit, minmax(300px, 1fr));
    gap: 25px;
    margin-top: 30px;
}
```

```
.практический-пример {
    background: white;
    padding: 25px;
    border-radius: 15px;
    border: 2px solid #e9ecf;
}

.практический-пример h3 {
    color: #2c3e50;
    margin-bottom: 20px;
    display: flex;
    align-items: center;
    gap: 10px;
}

.пример-визуальный {
    padding: 30px;
    border-radius: 10px;
    margin: 20px 0;
    text-align: center;
    min-height: 150px;
    display: flex;
    align-items: center;
    justify-content: center;
}

/* ШПАРГАЛКА */
.шпаргалка {
    background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);
    color: white;
    padding: 30px;
    border-radius: 20px;
    margin-top: 40px;
}

.шпаргалка h2 {
    color: white;
    border-bottom-color: rgba(255,255,255,0.3);
}

.карточки-шпаргалки {
    display: grid;
    grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));
```

```

        gap: 20px;
        margin-top: 25px;
    }

    .карточка-шпаргалка {
        background: rgba(255,255,255,0.1);
        padding: 25px;
        border-radius: 10px;
        backdrop-filter: blur(10px);
    }

    .карточка-шпаргалка h3 {
        color: white;
        margin-bottom: 15px;
        display: flex;
        align-items: center;
        gap: 10px;
    }

    /* АНИМАЦИЯ */
    @keyframes рамка-пульсация {
        0% { box-shadow: 0 0 0 rgba(52, 152, 219, 0.7); }
        70% { box-shadow: 0 0 0 10px rgba(52, 152, 219, 0); }
        100% { box-shadow: 0 0 0 rgba(52, 152, 219, 0); }
    }

    .пульсация {
        animation: рамка-пульсация 1.5s infinite;
    }

```

</style>

</head>

<body>

<h1> Лаборатория рамок: Искусство границ в CSS</h1>

<!-- Раздел 1: Что такое border -->

<div class="раздел">

<h2>1. Что такое border и зачем он нужен?</h2>

<div class="инфо-блок">

<p><strong>Border (граница, рамка)</strong> — это свойство CSS, которое создаёт рамку вокруг любого элемента на странице.</p>

<p>Представьте, что border — это <strong>багет для картины</strong> или <strong>рамка для фотографии</strong>. Он выполняет три функции:</p>

<ol>

```
<li><strong>Декоративная</strong> — делает элемент красивее</li>
<li><strong>Разделительная</strong> — отделяет один элемент от другого</li>
<li><strong>Акцентная</strong> — привлекает внимание к важному
```

элементу</li>

```
</ol>
```

```
</div>
```

```
<div style="text-align: center; margin: 30px 0;">
  <div style="display: inline-block; padding: 30px; background: white; border-radius: 10px;">
    <div style="font-size: 24px; color: #3498db; margin-bottom: 15px;">border: 3px solid red;</div>
    <div style="display: flex; justify-content: center; gap: 20px; font-size: 18px; color: #666;">
      <div>3px ← <strong>толщина</strong></div>
      <div>solid ← <strong>стиль</strong></div>
      <div>red ← <strong>цвет</strong></div>
    </div>
  </div>
</div>
</div>
```

<!-- Раздел 2: Галерея всех стилей рамок -->

```
<div class="раздел">
```

```
  <h2>2. Все стили рамок (border-style)</h2>
```

<p>CSS предлагает 8 разных стилей рамок. Выберите тот, который лучше всего подходит для вашей задачи:</p>

```
<div class="галерея-рамок">
```

```
  <!-- Solid -->
```

```
  <div class="рамка-пример" onclick="применитьСтиль('solid')">
```

```
    <div class="название-рамки">solid (сплошная)</div>
```

```
    <div style="width:100%; height:100px; border: 5px solid #3498db; border-radius: 5px; display: flex; align-items: center; justify-content: center; background: #f8f9fa;">
```

Самый частый стиль

```
</div>
```

```
  <div class="код-рамки">border: 5px solid #3498db;</div>
```

```
</div>
```

```
<!-- Dotted -->
```

```
  <div class="рамка-пример" onclick="применитьСтиль('dotted')">
```

```
    <div class="название-рамки">dotted (пунктирная)</div>
```

```
    <div style="width:100%; height:100px; border: 5px dotted #e74c3c; border-radius: 5px; display: flex; align-items: center; justify-content: center; background: #f8f9fa;">
```

Как будто точки

```
</div>
<div class="код-рамки">border: 5px dotted #e74c3c;</div>
</div>

<!-- Dashed -->
<div class="рамка-пример" onclick="применитьСтиль('dashed')">
    <div class="название-рамки">dashed (тире)</div>
    <div style="width:100%; height:100px; border: 5px dashed #2ecc71; border-radius: 5px; display: flex; align-items: center; justify-content: center; background: #f8f9fa;">
```

Короткие чёрточки

```
</div>
<div class="код-рамки">border: 5px dashed #2ecc71;</div>
</div>

<!-- Double -->
<div class="рамка-пример" onclick="применитьСтиль('double')">
    <div class="название-рамки">double (двойная)</div>
    <div style="width:100%; height:100px; border: 10px double #f39c12; border-radius: 5px; display: flex; align-items: center; justify-content: center; background: #f8f9fa;">
```

Две линии рядом

```
</div>
<div class="код-рамки">border: 10px double #f39c12;</div>
</div>

<!-- Groove -->
<div class="рамка-пример" onclick="применитьСтиль('groove')">
    <div class="название-рамки">groove (объёмная)</div>
    <div style="width:100%; height:100px; border: 10px groove #9b59b6; border-radius: 5px; display: flex; align-items: center; justify-content: center; background: #f8f9fa;">
```

Кажется вдавленной

```
</div>
<div class="код-рамки">border: 10px groove #9b59b6;</div>
</div>

<!-- Ridge -->
<div class="рамка-пример" onclick="применитьСтиль('ridge')">
    <div class="название-рамки">ridge (рельефная)</div>
    <div style="width:100%; height:100px; border: 10px ridge #1abc9c; border-radius: 5px; display: flex; align-items: center; justify-content: center; background: #f8f9fa;">
```

Кажется выпуклой

```
</div>
<div class="код-рамки">border: 10px ridge #1abc9c;</div>
</div>
```

```

<!-- Inset -->
<div class="рамка-пример" onclick="применитьСтиль('inset')">
    <div class="название-рамки">Inset (вставленная)</div>
    <div style="width:100%; height:100px; border: 10px inset #34495e; border-radius: 5px; display: flex; align-items: center; justify-content: center; background: #f8f9fa;">
        Как будто внутри
    </div>
    <div class="код-рамки">border: 10px inset #34495e;</div>
</div>

<!-- Outset -->
<div class="рамка-пример" onclick="применитьСтиль('outset')">
    <div class="название-рамки">Outset (выступающая)</div>
    <div style="width:100%; height:100px; border: 10px outset #e67e22; border-radius: 5px; display: flex; align-items: center; justify-content: center; background: #f8f9fa;">
        Как будто снаружи
    </div>
    <div class="код-рамки">border: 10px outset #e67e22;</div>
</div>
</div>

<!-- Раздел 3: Интерактивная лаборатория -->
<div class="раздел">
    <h2>3. Создайте свою уникальную рамку!</h2>
    <p>Поэкспериментируйте с настройками и посмотрите, как меняется рамка в реальном времени:</p>

    <div class="интерактивная-зона">
        <!-- Предпросмотр -->
        <div>
            <h3 style="color: #2c3e50; margin-bottom: 20px;">Предпросмотр вашей
рамки:</h3>
            <div class="предпросмотр">
                <div class="образец-рамки" id="образецРамки">
                    <h3>Ваша фотография</h3>
                    <p>или другой контент</p>
                    <div style="margin-top: 15px; font-size: 14px; color: #666;">
                        Эта рамка создана вами!
                    </div>
                </div>
            </div>
        </div>
    </div>

```

```
<div style="margin-top: 20px; text-align: center;">
    <div id="кодВашейРамки" style="background: #2c3e50; color: white; padding:
15px; border-radius: 8px; font-family: 'Courier New', monospace; font-size: 16px;">
        border: 5px solid #3498db;
    </div>
    <button onclick="скопироватьКод()" style="margin-top: 10px; background:
#27ae60; color: white; border: none; padding: 10px 20px; border-radius: 5px; cursor: pointer;">
        Скопировать код рамки
    </button>
</div>
</div>

<!-- Панель управления --&gt;
&lt;div class="панель-управления"&gt;
    &lt;div class="регуляторы"&gt;
        &lt;!-- Толщина рамки --&gt;
        &lt;div class="регулятор-группа"&gt;
            &lt;h3&gt;&lt;i class="fas fa-ruler-horizontal"&gt;&lt;/i&gt; Толщина рамки&lt;/h3&gt;
            &lt;label&gt;
                Толщина: &lt;span id="толщинаЗначение"
class="значение"&gt;5px&lt;/span&gt;
                &lt;input type="range" class="ползунок" id="толщинаРамки" min="1"
max="50" value="5" oninput="обновитьРамку()"&gt;
            &lt;/label&gt;
            &lt;div style="display: flex; justify-content: space-between; margin-top: 10px;
color: #666; font-size: 14px;"&gt;
                &lt;span&gt;Тонкая&lt;/span&gt;
                &lt;span&gt;Средняя&lt;/span&gt;
                &lt;span&gt;Толстая&lt;/span&gt;
            &lt;/div&gt;
        &lt;/div&gt;
    &lt;/div&gt;

    &lt;!-- Стиль рамки --&gt;
    &lt;div class="регулятор-группа"&gt;
        &lt;h3&gt;&lt;i class="fas fa-brush"&gt;&lt;/i&gt; Стиль рамки&lt;/h3&gt;
        &lt;div class="стили-рамок"&gt;
            &lt;button class="кнопка-стиль active" data-стиль="solid"
onclick="выбратьСтиль('solid')"&gt;Solid&lt;/button&gt;
            &lt;button class="кнопка-стиль" data-стиль="dotted"
onclick="выбратьСтиль('dotted')"&gt;Dotted&lt;/button&gt;
            &lt;button class="кнопка-стиль" data-стиль="dashed"
onclick="выбратьСтиль('dashed')"&gt;Dashed&lt;/button&gt;
            &lt;button class="кнопка-стиль" data-стиль="double"
onclick="выбратьСтиль('double')"&gt;Double&lt;/button&gt;
        &lt;/div&gt;
    &lt;/div&gt;
&lt;/div&gt;</pre>
```

```
<button class="кнопка-стиль" data-стиль="groove"
onclick="выбратьСтиль('groove')>Groove</button>
<button class="кнопка-стиль" data-стиль="ridge"
onclick="выбратьСтиль('ridge')>Ridge</button>
</div>
</div>

<!-- Цвет рамки -->
<div class="регулятор-группа">
<h3><i class="fas fa-palette"></i> Цвет рамки</h3>
<div class="цвета-рамок">
<div class="кнопка-цвет active" style="background: #3498db;" data-
цвет="#3498db" onclick="выбратьЦвет('#3498db')></div>
<div class="кнопка-цвет" style="background: #e74c3c;" data-
цвет="#e74c3c" onclick="выбратьЦвет('#e74c3c')></div>
<div class="кнопка-цвет" style="background: #2ecc71;" data-
цвет="#2ecc71" onclick="выбратьЦвет('#2ecc71')></div>
<div class="кнопка-цвет" style="background: #f39c12;" data-
цвет="#f39c12" onclick="выбратьЦвет('#f39c12')></div>
<div class="кнопка-цвет" style="background: #9b59b6;" data-
цвет="#9b59b6" onclick="выбратьЦвет('#9b59b6')></div>
<div class="кнопка-цвет" style="background: #1abc9c;" data-
цвет="#1abc9c" onclick="выбратьЦвет('#1abc9c')></div>
<div class="кнопка-цвет" style="background: #34495e;" data-
цвет="#34495e" onclick="выбратьЦвет('#34495e')></div>
<div class="кнопка-цвет" style="background: #e67e22;" data-
цвет="#e67e22" onclick="выбратьЦвет('#e67e22')></div>
<div class="кнопка-цвет" style="background: #7f8c8d;" data-
цвет="#7f8c8d" onclick="выбратьЦвет('#7f8c8d')></div>
<div class="кнопка-цвет" style="background: #d35400;" data-
цвет="#d35400" onclick="выбратьЦвет('#d35400')></div>
</div>
<div style="margin-top: 15px;">
<label>
    Свой цвет:
    <input type="color" id="выборЦвета" value="#3498db"
 onchange="выбратьСвойЦвет(this.value)" style="width: 50px; height: 30px; vertical-align: middle;
 margin-left: 10px;">
</label>
</div>
</div>

<!-- Скругление углов -->
<div class="регулятор-группа">
```

```

        <h3><i class="fas fa-circle-notch"></i> Скругление углов</h3>
        <label>
            Радиус: <span id="радиусЗначение">
class="значение">10px</span>
            <input type="range" class="ползунок" id="радиусУглов" min="0"
max="50" value="10" oninput="обновитьРамку()">
        </label>
        <div style="display: flex; justify-content: space-between; margin-top: 10px;
color: #666; font-size: 14px;">
            <span>Острые</span>
            <span>Скруглённые</span>
            <span>Очень круглые</span>
        </div>
    </div>
</div>


<div class="кнопки-действий">
    <button class="кнопка-действия" onclick="сохранитьРамку()">
        <i class="fas fa-save"></i> Сохранить рамку
    </button>
    <button class="кнопка-действия" onclick="сброситьРамку()">
        <i class="fas fa-redo"></i> Сбросить
    </button>
    <button class="кнопка-действия" onclick="поделитьсяРамкой()">
        <i class="fas fa-share-alt"></i> Поделиться
    </button>
</div>
</div>
</div>

<!-- Раздел 4: Практические примеры --&gt;
&lt;div class="раздел"&gt;
    &lt;h2&gt;4. Практическое применение рамок&lt;/h2&gt;
    &lt;p&gt;Вот как можно использовать рамки в реальных проектах:&lt;/p&gt;

    &lt;div class="примеры-практические"&gt;
        &lt;!-- Пример 1: Кнопка --&gt;
        &lt;div class="практический-пример"&gt;
            &lt;h3&gt;&lt;i class="fas fa-mouse-pointer"&gt;&lt;/i&gt; Кнопка с рамкой&lt;/h3&gt;
            &lt;div class="пример-визуальный" style="background: #f8f9fa;"&gt;
                &lt;button style="padding: 15px 30px; background: white; border: 3px solid
#3498db; border-radius: 8px; color: #3498db; font-size: 16px; cursor: pointer; transition: all 0.3s;"&gt;
</pre>

```

```

        onmouseover="this.style.background='#3498db';
this.style.color='white'";
        onmouseout="this.style.background='white';
this.style.color='#3498db">
    Нажми меня!
</button>
</div>
<div class="код-рамки" style="margin-top: 15px;">
    button {<br>
        &nbsp;&nbsp; border: 3px solid #3498db;<br>
        &nbsp;&nbsp; border-radius: 8px;<br>
        &nbsp;&nbsp; transition: all 0.3s;<br>
    }<br>
    button:hover {<br>
        &nbsp;&nbsp; background: #3498db;<br>
        &nbsp;&nbsp; color: white;<br>
    }
</div>
</div>

<!-- Пример 2: Карточка -->
<div class="практический-пример">
    <h3><i class="fas fa-id-card"></i> Карточка с акцентом</h3>
    <div class="пример-визуальный" style="background: white; border-left: 10px solid #2ecc71; padding: 25px; text-align: left;">
        <h4 style="margin: 0 0 10px 0; color: #2c3e50;">Важное объявление</h4>
        <p style="margin: 0; color: #666;">Рамка слева привлекает внимание и
делает карточку заметной.</p>
    </div>
    <div class="код-рамки" style="margin-top: 15px;">
        .карточка {<br>
            &nbsp;&nbsp; border-left: 10px solid #2ecc71;<br>
            &nbsp;&nbsp; padding: 25px;<br>
            &nbsp;&nbsp; background: white;<br>
        }
    </div>
</div>

<!-- Пример 3: Фото в рамке -->
<div class="практический-пример">
    <h3><i class="fas fa-image"></i> Фотография в рамке</h3>
    <div class="пример-визуальный" style="background: #f8f9fa; display: flex; align-items: center; justify-content: center;">

```

```
<div style="width: 150px; height: 150px; border: 15px solid #d4a762;  
background: white; display: flex; align-items: center; justify-content: center; font-size: 40px; color:  
#d4a762;">
```



```
</div>  
</div>  
<div class="код-рамки" style="margin-top: 15px;">  
.фото-рамка {<br>  
  &nbsp;&nbsp;border: 15px solid #d4a762;<br>  
  &nbsp;&nbsp;width: 150px;<br>  
  &nbsp;&nbsp;height: 150px;<br>  
  &nbsp;&nbsp;background: white;<br>  
}  
</div>  
</div>  
</div>
```

<!-- Шпаргалка -->

```
<div class="шпаргалка">
```

```
  <h2>📋 Шпаргалка: Всё о border</h2>
```

```
<div class="карточки-шпаргалки">
```

```
  <div class="карточка-шпаргалка">
```

```
    <h3><i class="fas fa-sliders-h"></i> Основные свойства</h3>
```

```
    <ul style="list-style: none; padding: 0;">
```

```
      <li style="margin-bottom: 10px;"><strong>border-width</strong> —  
толщина</li>
```

```
      <li style="margin-bottom: 10px;"><strong>border-style</strong> — стиль
```

```
линий</li>
```

```
      <li style="margin-bottom: 10px;"><strong>border-color</strong> — цвет</li>
```

```
      <li><strong>border-radius</strong> — скругление углов</li>
```

```
    </ul>
```

```
  </div>
```

```
<div class="карточка-шпаргалка">
```

```
  <h3><i class="fas fa-palette"></i> Популярные комбинации</h3>
```

```
  <div style="margin-bottom: 10px;">
```

```
    <div style="display: inline-block; width: 15px; height: 15px; background:  
#3498db; border-radius: 3px; margin-right: 10px;"></div>
```

```
    <code>1px solid #3498db</code>
```

```
  </div>
```

```
  <div style="margin-bottom: 10px;">
```

```

<div style="display: inline-block; width: 15px; height: 15px; background:
#e74c3c; border-radius: 3px; margin-right: 10px;"></div>
    <code>3px dotted #e74c3c</code>
</div>
<div style="margin-bottom: 10px;">
    <div style="display: inline-block; width: 15px; height: 15px; background:
#2ecc71; border-radius: 3px; margin-right: 10px;"></div>
        <code>2px dashed #2ecc71</code>
    </div>
    <div>
        <div style="display: inline-block; width: 15px; height: 15px; background:
#f39c12; border-radius: 3px; margin-right: 10px;"></div>
            <code>5px double #f39c12</code>
        </div>
    </div>

<div class="карточка-шпаргалка">
    <h3><i class="fas fa-lightbulb"></i> Полезные советы</h3>
    <ul style="list-style: none; padding: 0;">
        <li style="margin-bottom: 10px;">Используйте <strong>border-
radius</strong> для скругления</li>
        <li style="margin-bottom: 10px;"><strong>box-shadow</strong> вместо
толстого border для теней</li>
        <li style="margin-bottom: 10px;">Прозрачные границы: <code>border-color:
transparent;</code></li>
        <li>Убирайте рамку: <code>border: none;</code></li>
    </ul>
</div>
</div>

<div style="text-align: center; margin-top: 30px;">
    <div style="background: rgba(255,255,255,0.2); padding: 15px; border-radius: 10px;
display: inline-block;">
        <strong>💡 Золотое правило:</strong> Рамка должна дополнять контент, а не
отвлекать от него.
        Если рамка слишком броская, пользователи будут смотреть на неё, а не на
содержимое.
    </div>
</div>
</div>

<!-- Подключаем Font Awesome для иконок -->
<link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/
all.min.css">

```

```
<script>

// =====
// ПЕРЕМЕННЫЕ ДЛЯ УПРАВЛЕНИЯ РАМКОЙ
// =====

let текущийСтиль = 'solid';
let текущийЦвет = '#3498db';
let текущаяТолщина = 5;
let текущийРадиус = 10;

// =====
// ФУНКЦИИ УПРАВЛЕНИЯ
// =====

// Обновление рамки при изменении параметров
function обновитьРамку() {
    const образец = document.getElementById('образецРамки');
    const толщина = document.getElementById('толщинаРамки').value;
    const радиус = document.getElementById('радиусУглов').value;

    // Обновляем значения
    текущаяТолщина = толщина;
    текущийРадиус = радиус;

    // Обновляем отображение значений
    document.getElementById('толщинаЗначение').textContent = толщина + 'px';
    document.getElementById('радиусЗначение').textContent = радиус + 'px';

    // Применяем стили к образцу
    образец.style.borderWidth = толщина + 'px';
    образец.style.borderStyle = текущийСтиль;
    образец.style.borderColor = текущийЦвет;
    образец.style.borderRadius = радиус + 'px';

    // Обновляем отображаемый код
    обновитьКодРамки();

    // Анимация пульсации
    образец.classList.add('пульсация');
    setTimeout(() => образец.classList.remove('пульсация'), 1500);
}

// Выбор стиля рамки
function выбратьСтиль(стиль) {
```

```
текущийСтиль = стиль;

// Обновляем активную кнопку
document.querySelectorAll('.кнопка-стиль').forEach(кнопка => {
    кнопка.classList.remove('active');
    if (кнопка.getAttribute('data-стиль') === стиль) {
        кнопка.classList.add('active');
    }
});

обновитьРамку();

}

// Выбор цвета из палитры
function выбратьЦвет(цвет) {
    текущийЦвет = цвет;

    // Обновляем активную кнопку
    document.querySelectorAll('.кнопка-цвет').forEach(кнопка => {
        кнопка.classList.remove('active');
        if (кнопка.getAttribute('data-цвет') === цвет) {
            кнопка.classList.add('active');
        }
    });

    // Обновляем цветовой инпут
    document.getElementById('выборЦвета').value = цвет;

    обновитьРамку();
}

// Выбор своего цвета
function выбратьСвойЦвет(цвет) {
    выбратьЦвет(цвет);
}

// Применение стиля из галереи
function применитьСтиль(стиль) {
    выбратьСтиль(стиль);

    // Автоматически подбираем цвет под стиль
    const цветаДляСтилей = {
        'solid': '#3498db',
        'dotted': '#e74c3c',
    }
}
```

```

'dashed': '#2ecc71',
'double': '#f39c12',
'groove': '#9b59b6',
'ridge': '#1abc9c',
'inset': '#34495e',
'outset': '#e67e22'
};

if (цветаДляСтилей[стиль]) {
    выбратьЦвет(цветаДляСтилей[стиль]);
}

// Обновление отображаемого кода
function обновитьКодРамки() {
    const кодЭлемент = document.getElementById('кодВашейРамки');
    const код = `border: ${текущаяТолщина}px ${текущийСтиль} ${текущийЦвет};\nborder-radius: ${текущийРадиус}px;`;
    кодЭлемент.innerHTML = код.replace(/\n/g, '<br>');
}

// Копирование кода рамки
function скопироватьКод() {
    const код = `border: ${текущаяТолщина}px ${текущийСтиль} ${текущийЦвет};\nborder-radius: ${текущийРадиус}px;`;

    navigator.clipboard.writeText(код)
        .then(() => {
            alert('Код рамки скопирован в буфер обмена!\n\n' + код);
        })
        .catch(err => {
            // Альтернативный способ для старых браузеров
            const textArea = document.createElement('textarea');
            textArea.value = код;
            document.body.appendChild(textArea);
            textArea.select();
            document.execCommand('copy');
            document.body.removeChild(textArea);
            alert('Код рамки скопирован!');
        });
}

// Сохранение рамки (в данном случае просто показ сообщения)
function сохранитьРамку() {

```

```
    alert('Ваша рамка сохранена!\n\nВы можете использовать этот код в своих
проектах:\n\n' +
      `border: ${текущаяТолщина}px ${текущийСтиль} ${текущийЦвет};\n` +
      `border-radius: ${текущийРадиус}px;`);
}

// Сброс рамки к начальным значениям
function сброситьРамку() {
  текущийСтиль = 'solid';
  текущийЦвет = '#3498db';

  document.getElementById('толщинаРамки').value = 5;
  document.getElementById('радиусУглов').value = 10;

  // Сбрасываем активные кнопки
  document.querySelectorAll('.кнопка-стиль').forEach(кнопка => {
    кнопка.classList.remove('active');
    if (кнопка.getAttribute('data-стиль') === 'solid') {
      кнопка.classList.add('active');
    }
  });

  document.querySelectorAll('.кнопка-цвет').forEach(кнопка => {
    кнопка.classList.remove('active');
    if (кнопка.getAttribute('data-цвет') === '#3498db') {
      кнопка.classList.add('active');
    }
  });

  document.getElementById('выборЦвета').value = '#3498db';

  обновитьРамку();
}

// Поделиться рамкой
function поделитьсяРамкой() {
  const текст = `Посмотрите рамку, которую я создал(а) в Лаборатории рамок!\n\n` +
    `Стиль: ${текущийСтиль}\n` +
    `Цвет: ${текущийЦвет}\n` +
    `Толщина: ${текущаяТолщина}px\n` +
    `Скругление: ${текущийРадиус}px\n\n` +
    `CSS-код:\n` +
    `border: ${текущаяТолщина}px ${текущийСтиль} ${текущийЦвет};\n` +
    `border-radius: ${текущийРадиус}px;`;
}
```

```

`border-radius: ${текущийРадиус}px;`


if (navigator.share) {
    navigator.share({
        title: 'Моя уникальная рамка',
        text: текст,
        url: window.location.href
    });
} else {
    // Альтернатива для браузеров без поддержки Web Share API
    скопироватьКод();
    alert('Код рамки скопирован! Поделитесь им с друзьями.');
}

// =====
// ИНИЦИАЛИЗАЦИЯ
// =====

window.onload = function() {
    // Инициализируем рамку с начальными значениями
    обновитьРамку();

    // Делаем образец рамки кликабельным для быстрого применения
    document.getElementById('образецРамки').addEventListener('click', function() {
        скопироватьКод();
    });

    // Добавляем подсказку
    document.getElementById('образецРамки').title = 'Кликните, чтобы скопировать код рамки';
};

</script>
</body>
</html>

```

---

## 4. Практические примеры для пенсионеров

### Пример 1: Рамка для фото в семейном альбоме

html

```
<!DOCTYPE html>
<html>
<head>
<style>

.фото-альбом {
    display: flex;
    flex-wrap: wrap;
    gap: 20px;
    padding: 30px;
    background: #f9f5f0;
}

.фото-рамка {
    width: 200px;
    height: 250px;
    padding: 15px;
    background: white;
    display: flex;
    flex-direction: column;
    align-items: center;
    justify-content: center;
    text-align: center;
    box-shadow: 5px 5px 15px rgba(0,0,0,0.1);
}

/* Разные стили рамок для разных фото */
.старая-фото {
    border: 10px double #8b4513; /* Двойная коричневая рамка */
    border-radius: 5px;
}

.цветная-фото {
    border: 15px solid transparent;
    border-image: linear-gradient(45deg, #ff9a9e, #fad0c4, #a1c4fd) 1;
    border-radius: 10px;
}

.детская-фото {
    border: 8px dotted #ff6b6b; /* Красная пунктирная рамка */
    border-radius: 20px;
    background: #fff9e6;
}

.современная-фото {
```

```
border: 3px solid #2c3e50; /* Тонкая тёмная рамка */
border-radius: 0;
box-shadow: 0 10px 30px rgba(0,0,0,0.15);
}

.photo-sign {
    margin-top: 10px;
    font-style: italic;
    color: #666;
    font-size: 14px;
}

```

</style>

</head>

<body>

<div class="photo-album">

<div class="photo-frame-old-photo">

<div style="width:150px; height:180px; background:#e3d5c5; display: flex; align-items: center; justify-content: center; font-size: 40px;">

相机

</div>

<div class="photo-sign">Бабушка, 1950 г.</div>

</div>

<div class="photo-frame-flower-photo">

<div style="width:150px; height:180px; background:#a1c4fd; display: flex; align-items: center; justify-content: center; font-size: 40px;">

粉色花束

</div>

<div class="photo-sign">Сад в цвету</div>

</div>

<div class="photo-frame-child-photo">

<div style="width:150px; height:180px; background:#ffebee; display: flex; align-items: center; justify-content: center; font-size: 40px;">

小孩

</div>

<div class="photo-sign">Внуки на даче</div>

</div>

<div class="photo-frame-modern-photo">

<div style="width:150px; height:180px; background:#f8f9fa; display: flex; align-items: center; justify-content: center; font-size: 40px;">

蛋糕

</div>

```
<div class="фото-подпись">Юбилей, 2023 г.</div>
</div>
</div>
</body>
</html>
```

## Пример 2: Кулинарные карточки с рецептами

```
html
<!DOCTYPE html>
<html>
<head>
<style>
.рецепты {
    max-width: 800px;
    margin: 0 auto;
    padding: 30px;
    background: #ffffaf0;
}

.рецепт-карточка {
    background: white;
    margin-bottom: 25px;
    border-radius: 15px;
    overflow: hidden;
    box-shadow: 0 5px 15px rgba(0,0,0,0.1);
}

/* Рамки для разных типов рецептов */
.рецепт-выпечка {
    border-left: 10px solid #d4a762; /* Золотистая рамка слева */
}

.рецепт-салат {
    border-left: 10px solid #27ae60; /* Зелёная рамка */
}

.рецепт-суп {
    border-left: 10px solid #e67e22; /* Оранжевая рамка */
}

.рецепт-заготовки {
    border-left: 10px dashed #8b4513; /* Коричневая пунктирная */
```

```
}

.рецепт-заголовок {
    background: #f8f9fa;
    padding: 20px;
    margin: 0;
    color: #2c3e50;
    border-bottom: 2px solid #e9ecf;
}

.рецепт-содержимое {
    padding: 25px;
    line-height: 1.6;
    color: #555;
}

.рецепт-совет {
    background: #fff8e1;
    padding: 15px;
    margin: 15px 0;
    border: 2px dashed #f39c12; /* Пунктирная жёлтая рамка */
    border-radius: 8px;
    font-style: italic;
}

.рецепт-ингредиенты {
    background: #f0f8ff;
    padding: 20px;
    border: 1px solid #b3d9ff;
    border-radius: 8px;
    margin: 20px 0;
}

</style>
</head>
<body>
<div class="рецепты">
    <div class="рецепт-карточка рецепт-выпечка">
        <h2 class="рецепт-заголовок">🍎 Яблочный пирог «Бабушкин»</h2>
        <div class="рецепт-содержимое">
            <div class="рецепт-ингредиенты">
                <strong>Ингредиенты:</strong> мука, яблоки, сахар, масло, яйца, корица
            </div>
            <p>Тесто замесить, раскатать, выложить яблоки, посыпать корицей, выпекать 40 минут при 180°C.</p>
        </div>
    </div>
</div>
```

```
<div class="рецепт-совет">
    <strong>Секрет бабушки:</strong> Добавьте щепотку лимонной цедры!
</div>
</div>

<div class="рецепт-карточка рецепт-салат">
    <h2 class="рецепт-заголовок">🥗 Винегрет «Дачный»</h2>
    <div class="рецепт-содержимое">
        <div class="рецепт-ингредиенты">
            <strong>Ингредиенты:</strong> свёкла, картофель, морковь, огурцы,
            горошек, лук, масло
        </div>
        <p>Овощи отварить, нарезать кубиками, смешать, заправить маслом.</p>
    </div>
</div>
</body>
</html>
```

---

## 5. Важные нюансы и продвинутые техники

### A. Разные рамки для разных сторон

css

```
.разные-стороны {
    border-top: 3px solid red; /* Только сверху */
    border-right: 2px dashed blue; /* Только справа */
    border-bottom: 5px dotted green; /* Только снизу */
    border-left: 4px double purple; /* Только слева */
}

/* Или сокращённая запись для всех сторон */
.все-стороны-разные {
    border-width: 1px 2px 3px 4px; /* top right bottom left */
    border-style: solid dashed dotted double;
    border-color: red blue green yellow;
}
```

## В. Скругление углов (border-radius)

css

```
/* Простое скругление */
.скругленные {
    border-radius: 10px; /* Все углы одинаково */
}

/* Разное скругление для разных углов */
.разные-углы {
    border-radius: 10px 20px 30px 40px; /* top-left top-right bottom-right bottom-left */
}

/* Эллиптическое скругление */
.эллиптическое {
    border-radius: 50px / 25px; /* горизонтальный радиус / вертикальный радиус */
}

/* Круг (для аватарки) */
.круг {
    border-radius: 50%;
    border: 3px solid gold;
}
```

## С. Градиентные рамки (продвинутая техника)

css

```
.градиентная-рамка {
    border: 5px solid transparent;
    border-image: linear-gradient(45deg, #ff9a9e, #fad0c4, #a1c4fd) 1;
    border-radius: 10px;
}

/* Или с градиентом только по одной стороне */
.верх-градиент {
    border-top: 5px solid transparent;
    border-image: linear-gradient(to right, #667eea, #764ba2) 1;
}
```

## Д. Прозрачные и невидимые рамки

css

```
/* Прозрачная рамка (занимает место, но не видна) */
.прозрачная {
    border: 3px solid transparent;
}

/* Рамка появляется только при наведении */
.при-наведении {
    border: 2px solid transparent;
    transition: border-color 0.3s;
}

.при-наведении:hover {
    border-color: #3498db;
}

/* Убираем рамку вообще */
.без-рамки {
    border: none;
    outline: none; /* Убираем и outline (рамку фокуса) */
}
```

## E. Outline vs Border

css

```
/* Border занимает место в макете */
.c-border {
    border: 3px solid red;
    /* Ширина элемента увеличится на 6px (3px слева + 3px справа) */
}

/* Outline не занимает место, рисуется поверх */
.c-outline {
    outline: 3px solid blue;
    /* Ширина элемента не изменится */
}
```

---

## 6. Итог: Простые правила для пенсионеров

### Базовые комбинации, которые всегда работают:

css

```
/* Для кнопок */
.кнопка {
    border: 2px solid #3498db;
    border-radius: 5px;
}

/* Для карточек */
.карточка {
    border: 1px solid #ddd;
    border-radius: 10px;
}

/* Для фотографий */
.фото {
    border: 5px solid white;
    border-radius: 3px;
    box-shadow: 0 3px 10px rgba(0,0,0,0.2);
}

/* Для важных сообщений */
.важно {
    border-left: 5px solid #e74c3c;
    background: #ffebee;
}

/* Для красивых заголовков */
.красивый-заголовок {
    border-bottom: 3px solid #9b59b6;
    padding-bottom: 10px;
}
```

### Золотые правила:

1. **Начинайте с тонких рамок** (1-3px), толстые часто выглядят грубо
2. **Используйте border-radius** для скругления углов — выглядит современнее
3. **Цвет рамки должен сочетаться** с цветом текста или фона

4. **Не переусердствуйте** — одна-две рамки на странице достаточно
5. **Проверяйте контраст** — тёмная рамка на тёмном фоне не видна

## Когда что использовать:

- solid — для большинства случаев (кнопки, карточки)
- dashed/dotted — для черновиков, временных элементов
- double — для элегантного, классического вида
- groove/ridge — для объёмных, "трёхмерных" эффектов
- **Только с одной стороны** — для разделения контента или акцентов

## Простая памятка:

css

```
/* Самые полезные комбинации */  
.стандартная { border: 2px solid #ddd; border-radius: 5px; }  
.акцентная { border: 3px solid #3498db; border-radius: 8px; }  
.элегантная { border: 1px solid #8b4513; border-radius: 0; }  
.веселая { border: 4px dotted #e74c3c; border-radius: 15px; }  
.невидимая { border: 2px solid transparent; } /* Занимает место для будущей анимации */
```

**Теперь вы — мастер рамок!** Вы можете создавать красивые границы для любых элементов, от простых кнопок до сложных галерей. В следующей главе мы научимся добавлять тени к элементам — это сделает их объёмными и реалистичными! 

## ■ Свойство **box-shadow**: добавляем легкую тень, чтобы элемент «парил» над страницей.

Дорогой друг, помните, как в солнечный день предметы отбрасывают тени? Тень делает их объёмными, «живыми», отделяет от фона. В веб-дизайне мы можем создавать такие же тени с помощью волшебного свойства **box-shadow**. Это один из самых простых способов сделать вашу страницу современной, стильной и «объёмной».

---

### Что такое **box-shadow**?

**box-shadow** — это свойство CSS, которое добавляет тень к блочному элементу (к той самой «рамке для фотографии», о которой мы говорили). Тень может быть лёгкой и нежной или яркой и выразительной — всё зависит от вашей творческой задачи.

Простейшая запись выглядит так:

css

```
.моя-рамка {  
    box-shadow: 5px 5px 10px grey;  
}
```

Эта строчка говорит: «Создай тень, сдвинутую на 5px вправо и 5px вниз, размой её на 10px и сделай серого цвета».

---

### Разберём «рецепт тени» по шагам

У **box-shadow** есть несколько параметров, которые идут всегда в одном порядке. Давайте представим, что мы печём пирог — у нас есть чёткий рецепт:

text

box-shadow: [сдвиг по X] [сдвиг по Y] [размытие] [растяжение] [цвет] [внутренняя?];

## **1. Сдвиг по горизонтали (X)**

На сколько пикселей тень смещена вправо. Если число отрицательное (-5px) — тень сместится влево.

## **2. Сдвиг по вертикали (Y)**

На сколько пикселей тень смещена вниз. Если число отрицательное (-5px) — тень сместится вверх.

## **3. Размытие (необязательно, но очень важно!)**

Чем больше число, тем мягче и «воздушнее» тень. Если поставить 0, тень будет резкой, как от картонки.

## **4. Растворение (необязательно)**

На сколько пикселей увеличить или уменьшить тень. Редко используется, но иногда нужно.

## **5. Цвет (можно задавать любым способом — HEX, RGB, название)**

Цвет самой тени. Чаще всего используют полупрозрачный чёрный `rgba(0,0,0,0.3)`.

## **6. Внутренняя тень (необязательно)**

Ключевое слово `inset` сделает тень не снаружи элемента, а внутри, как бы «вдавленной».

---

# **Примеры тени в деле**

## **Пример 1: Лёгкая тень для «парения»**

css

```
.визитка {  
    box-shadow: 0 5px 15px rgba(0, 0, 0, 0.1);  
}
```

Эффект: Элемент слегка приподнят над страницей, как будто он лежит на столе и отбрасывает мягкую тень.

## Пример 2: Тень для кнопки при наведении

css

```
.кнопка {  
    box-shadow: 0 2px 5px rgba(0,0,0,0.2);  
}  
.кнопка:hover {  
    box-shadow: 0 5px 15px rgba(0,0,0,0.3);  
}
```

Эффект: При наведении мыши кнопка «поднимается» выше — это интерактивно и современно.

## Пример 3: Внутренняя тень для вдавленного эффекта

css

```
.поле-ввода {  
    box-shadow: inset 0 1px 3px rgba(0,0,0,0.2);  
}
```

Эффект: Поле кажется углублённым в страницу, как выгравированное.

## Пример 4: Множественные тени

css

```
.красивый-блок {  
    box-shadow:  
        0 2px 5px rgba(0,0,0,0.1),  
        0 5px 15px rgba(0,0,0,0.05);  
}
```

Можно задавать несколько теней через запятую! Первая будет ближе к элементу, вторая — дальше и размытее. Это создаёт сложный, красивый эффект.

---

## Практикум №4: Украшаем нашу визитку красивой рамкой и тенью

Давайте применим новые знания на практике. У нас уже есть визитка из предыдущего практикума. Сейчас мы сделаем её по-настоящему стильной!

**Шаг 1: Откройте ваш файл style.css (или создайте новый)**

**Шаг 2: Добавьте стили для визитки с тенью:**

css

```
/* Визитка с рамкой и тенью */
.визитка {
    width: 400px;
    background-color: white;
    padding: 30px;
    border-radius: 15px; /* Закругляем углы */
    border: 1px solid #f0f0f0; /* Очень тонкая светлая рамка */

    /* ВОТ ВОЛШЕБНАЯ ТЕНЬ! */
    box-shadow: 0 10px 30px rgba(0, 0, 0, 0.08);

    margin: 50px auto; /* Центрируем на странице */
    font-family: 'Georgia', serif;
}

.визитка:hover {
    /* При наведении тень становится чуть больше и темнее */
    box-shadow: 0 15px 40px rgba(0, 0, 0, 0.12);
    transform: translateY(-5px); /* Лёгкий подъём */
    transition: all 0.3s ease; /* Плавная анимация */
}

/* Заголовок визитки */
.визитка h2 {
    color: #2c3e50;
    text-align: center;
    margin-bottom: 10px;
    border-bottom: 2px solid #3498db;
    padding-bottom: 10px;
}
```

```

/* Контакты */
.контакты {
    background: #f8f9fa;
    padding: 15px;
    border-radius: 8px;
    margin-top: 20px;
    font-size: 16px;
}

/* Элемент в контактах с внутренней тенью */
.контакт-строка {
    padding: 8px;
    margin: 5px 0;
    border-radius: 4px;
    box-shadow: inset 0 1px 2px rgba(0,0,0,0.05);
}

```

### Шаг 3: Создайте HTML-файл vizitka.html:

```

html
<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Моя визитка с тенью</title>
    <link rel="stylesheet" href="style.css">
</head>
<body>
    <div class="визитка">
        <h2>Иванова Галина Петровна</h2>
        <p><strong>Садовод-любитель, мастер вязания</strong></p>
        <p>Опыт более 40 лет. Создаю уют своими руками и делаю теплом с близкими.</p>

        <div class="контакты">
            <div class="контакт-строка">📞 Телефон: +7 (912) 345-67-89</div>
            <div class="контакт-строка">✉ Email: galina.sadovod@example.com</div>
            <div class="контакт-строка">📍 Адрес: г. Екатеринбург, ул. Цветочная, 15</div>
        </div>

        <p style="text-align: center; margin-top: 20px; font-style: italic; color: #7f8c8d;">
            «Красота спасёт мир, если мир позволит ей это сделать»
        </p>
    </div>

```

```
</div>
</body>
</html>
```

## Шаг 4: Экспериментируйте с тенями!

Попробуйте изменить значения в `box-shadow` и посмотрите, как меняется визитка:

### 1. Увеличьте размытие:

css

`box-shadow: 0 10px 50px rgba(0, 0, 0, 0.1);`

### 2. Сделайте тень темнее:

css

`box-shadow: 0 10px 30px rgba(0, 0, 0, 0.2);`

### 3. Создайте тень только снизу:

css

`box-shadow: 0 10px 20px -5px rgba(0, 0, 0, 0.15);`

(Отрицательное растяжение «сжимает» тень)

### 4. Добавьте цветную тень:

css

`box-shadow: 0 10px 30px rgba(139, 69, 19, 0.2);`

(Коричневая тень для тёплого стиля)

---

## Важные советы по использованию теней

- Не переборщите!** Слишком много теней или слишком тёмные тени создают «грязь» на странице.
- Соблюдайте единый стиль.** Если вы используете тени на одном элементе, используйте похожие тени на других.
- Учитывайте источник света.** Обычно тень падает вниз и чуть вправо (как от солнца сверху слева).
- Для текста используйте** `text-shadow`. Это отдельное свойство для теней у текста!

**5. Экспериментируйте в инструментах разработчика (F12).** Меняйте значения в реальном времени и смотрите результат.

---

## Домашнее задание

### 1. Создайте «парящую» фотографию

Возьмите свою любимую фотографию (или картинку), добавьте ей белую рамку и лёгкую тень, чтобы она казалась лежащей на столе.

### 2. Сделайте набор «плавающих» карточек

3 карточки с разными рецептами (или хобби), расположенные рядом. У каждой должна быть своя тень, и при наведении они должны слегка подниматься.

### 3. Поэкспериментируйте с внутренними тенями

Создайте поле для поиска на сайте, которое выглядит вдавленным с помощью `box-shadow: inset;`

**Помните:** Тень — это как специя в блюде. Немного — вкусно и стильно, слишком много — можно испортить. Но не бойтесь экспериментировать! Ведь у нас всё безопасно, и любую тень можно легко убрать или изменить.

---

**Вот и всё! Теперь вы умеете создавать объёмные, «живые» элементы с помощью теней.** В следующей главе мы научимся управлять размером наших «рамок» — делать их шире, уже, выше или ниже. Это будет просто, ведь главное — понять основы!

## ■ Практикум №4: Украшаем нашу визитку красивой рамкой и тенью.

Дорогой друг! Пришло время превратить нашу простую визитку в настоящее произведение веб-дизайна. Мы научимся создавать элегантные рамки и добавлять лёгкие тени, которые заставят элементы «парить» над страницей. Это один из самых эффектных приёмов в CSS — простой, но очень мощный!

### 🎯 Цель практикума:

Создать профессионально выглядящую визитную карточку с:

- Аккуратной рамкой
  - Лёгкой объёмной тенью
  - Эффектом приподнимания при наведении
  - Закруглёнными углами
- 

### 📁 Подготовка рабочего пространства

#### Шаг 1: Создайте папку для проекта

На Рабочем столе создайте новую папку:

text

Моя\_Визитка\_С\_Тенью

#### Шаг 2: Создайте файлы

В папке создайте два файла:

1. index.html — структура визитки
  2. style.css — стили и оформление
-



## Шаг 3: Создаём HTML-структурку (index.html)

Откройте файл index.html в VS Code и вставьте следующий код:

```
html

<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Моя визитка с рамкой и тенью</title>
    <!-- Подключаем наш файл со стилями -->
    <link rel="stylesheet" href="style.css">
    <!-- Подключаем красивый шрифт из Google Fonts -->
    <link rel="preconnect" href="https://fonts.googleapis.com">
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
    <link href="https://fonts.googleapis.com/css2?
family=Playfair+Display:wght@400;700&family=Raleway:wght@300;400;600&display=swap"
rel="stylesheet">
</head>
<body>
    <div class="контейнер">
        <h1 class="заголовок-страницы">Моя Визитная Карточка</h1>
        <p class="подзаголовок">Наведите курсор на визитку, чтобы увидеть магию теней!</p>

        <!-- Основная визитка -->
        <div class="визитка" id="основная-визитка">
            <div class="верхняя-часть">
                <div class="аватар-контейнер">
                    <div class="аватар"></div>
                </div>
                <h2 class="имя">Иванова Галина Петровна</h2>
                <p class="должность">Садовод-любитель, мастер вязания</p>
            </div>

            <div class="разделитель"></div>

            <div class="информация">
                <div class="инфо-блок">
                    <div class="иконка">🌿</div>
                    <div class="текст">
                        <h3>Моё хобби</h3>
                    </div>
                </div>
            </div>
        </div>
    </div>
```

```
<p>Выращиваю розы и вяжу тёплые вещи для внуков. Опыт — более  
40 лет.</p>  
</div>  
</div>  
  
<div class="инфо-блок">  
    <div class="иконка">📞</div>  
    <div class="текст">  
        <h3>Контакты</h3>  
        <p>Телефон: +7 (912) 345-67-89</p>  
        <p>Email: galina.sad@example.com</p>  
    </div>  
</div>  
  
<div class="инфо-блок">  
    <div class="иконка">📍</div>  
    <div class="текст">  
        <h3>Адрес</h3>  
        <p>г. Екатеринбург, ул. Цветочная, 15</p>  
    </div>  
</div>  
</div>  
  
<div class="цитата">  
    <p>«Цветы не решают всех проблем, но они — прекрасное начало.»</p>  
</div>  
</div>  
  
<!-- Мини-галерея с разными вариантами теней -->  
<div class="галерея-теней">  
    <h3>Экспериментируйте с разными тенями:</h3>  
    <div class="тень-варианты">  
        <div class="вариант" id="лёгкая-тень">  
            <div class="предпросмотр-тени"></div>  
            <p>Лёгкая</p>  
        </div>  
        <div class="вариант" id="выраженная-тень">  
            <div class="предпросмотр-тени"></div>  
            <p>Выраженная</p>  
        </div>  
        <div class="вариант" id="цветная-тень">  
            <div class="предпросмотр-тени"></div>  
            <p>Цветная</p>  
        </div>  
    </div>
```

```
<div class="вариант" id="двойная-тень">
    <div class="предпросмотр-тени"></div>
    <p>Двойная</p>
</div>
</div>

<!-- Панель управления -->
<div class="панель-управления">
    <h3>🎨 Лаборатория теней</h3>

    <div class="управление-группа">
        <label>Сдвиг по X: <span id="значение-x">5</span>px</label>
        <input type="range" id="ползунок-x" min="0" max="20" value="5">
    </div>

    <div class="управление-группа">
        <label>Сдвиг по Y: <span id="значение-y">5</span>px</label>
        <input type="range" id="ползунок-y" min="0" max="20" value="5">
    </div>

    <div class="управление-группа">
        <label>Размытие: <span id="значение-размытие">15</span>px</label>
        <input type="range" id="ползунок-размытие" min="0" max="50" value="15">
    </div>

    <div class="управление-группа">
        <label>Прозрачность: <span id="значение-прозрачность">0.2</span></label>
        <input type="range" id="ползунок-прозрачность" min="0" max="100" value="20">
    </div>

    <div class="кнопки">
        <button id="кнопка-сброс" class="кнопка">Сбросить</button>
        <button id="кнопка-код" class="кнопка">Показать CSS-код</button>
    </div>
</div>

<div class="подсказка">
    <p>💡 <strong>Совет:</strong> Поменяйте значения с помощью ползунков и
    посмотрите, как меняется тень на визитке в реальном времени!</p>
</div>
</div>

<script src="скрипт.js"></script>
```

```
</body>  
</html>
```

---

## Шаг 4: Создаём стили (style.css)

Теперь создайте файл style.css и вставьте в него этот код:

css

```
/* ===== ОСНОВНЫЕ НАСТРОЙКИ СТРАНИЦЫ ===== */  
* {  
    box-sizing: border-box;  
    margin: 0;  
    padding: 0;  
}  
  
body {  
    font-family: 'Raleway', 'Arial', sans-serif;  
    line-height: 1.6;  
    color: #333;  
    background: linear-gradient(135deg, #f5f7fa 0%, #c3cfec 100%);  
    min-height: 100vh;  
    padding: 20px;  
}  
  
.контейнер {  
    max-width: 900px;  
    margin: 0 auto;  
    padding: 20px;  
}  
  
.заголовок-страницы {  
    font-family: 'Playfair Display', 'Georgia', serif;  
    text-align: center;  
    color: #2c3e50;  
    margin-bottom: 15px;  
    font-size: 2.5rem;  
    text-shadow: 1px 1px 3px rgba(0,0,0,0.1);  
}  
  
.подзаголовок {  
    text-align: center;
```

```
color: #7f8c8d;
margin-bottom: 40px;
font-size: 1.2rem;
}

/* ===== ОСНОВНАЯ ВИЗИТКА ===== */
.визитка {
    width: 100%;
    max-width: 600px;
    margin: 0 auto 50px;
    background-color: white;

    /* ЗАКРУГЛЁННЫЕ УГЛЫ */
    border-radius: 20px;

    /* РАМКА - простая, элегантная */
    border: 1px solid #eaeaea;

    /* ТЕНЬ ПО УМОЛЧАНИЮ - лёгкая, воздушная */
    box-shadow: 0 5px 15px rgba(0, 0, 0, 0.08);

    padding: 40px;
    position: relative;
    overflow: hidden;

    /* Плавный переход для всех изменений */
    transition: all 0.4s ease;
}

/* Эффект при наведении на визитку */
.визитка:hover {
    /* УСИЛЕННАЯ ТЕНЬ при наведении */
    box-shadow: 0 15px 35px rgba(0, 0, 0, 0.12);

    /* Лёгкий подъём */
    transform: translateY(-10px);

    /* Более выраженная рамка */
    border-color: #d5d5d5;
}

/* ===== ВЕРХНЯЯ ЧАСТЬ ВИЗИТКИ ===== */
.верхняя-часть {
    text-align: center;
```

```
margin-bottom: 30px;
}

.аватар-контейнер {
    width: 120px;
    height: 120px;
    margin: 0 auto 25px;
    border-radius: 50%;

    /* РАМКА для аватара */
    border: 4px solid #f8f9fa;

    /* ТЕНЬ для аватара */
    box-shadow: 0 4px 10px rgba(0,0,0,0.1);

    overflow: hidden;
    background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);
}

.аватар {
    width: 100%;
    height: 100%;
    background: url('https://images.unsplash.com/photo-1573496359142-b8d87734a5a2?ixlib=rb-1.2.1&auto=format&fit=crop&w=500&q=80') center/cover;
    opacity: 0.8;
}

.имя {
    font-family: 'Playfair Display', serif;
    font-size: 2rem;
    color: #2c3e50;
    margin-bottom: 10px;
}

.должность {
    color: #3498db;
    font-size: 1.2rem;
    font-weight: 600;
}

/* ===== РАЗДЕЛИТЕЛЬ ===== */
.разделитель {
    height: 1px;
    background: linear-gradient(to right, transparent, #e0e0e0, transparent);
}
```

```
margin: 30px 0;  
}  
  
/* ===== БЛОКИ ИНФОРМАЦИИ ===== */  
.информация {  
    display: grid;  
    grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));  
    gap: 25px;  
    margin-bottom: 30px;  
}  
  
.инфо-блок {  
    display: flex;  
    align-items: flex-start;  
    padding: 20px;  
    background-color: #f8f9fa;  
    border-radius: 12px;  
  
    /* ТЕНЬ ВНУТРИ блока (inset) */  
    box-shadow: inset 0 2px 5px rgba(0,0,0,0.05);  
  
    transition: all 0.3s ease;  
}  
  
.инфо-блок:hover {  
    background-color: #f0f7ff;  
    transform: translateY(-3px);  
  
    /* Внешняя тень при наведении */  
    box-shadow:  
        inset 0 2px 5px rgba(0,0,0,0.05),  
        0 5px 15px rgba(52, 152, 219, 0.1);  
}  
  
.иконка {  
    font-size: 2rem;  
    margin-right: 15px;  
}  
  
.текст h3 {  
    color: #2c3e50;  
    margin-bottom: 8px;  
    font-size: 1.3rem;  
}
```

```
.текст p {
    color: #555;
    font-size: 1rem;
    line-height: 1.5;
}

/* ===== ЦИТАТА ===== */

.цитата {
    padding: 25px;
    background: linear-gradient(135deg, #f093fb 0%, #f5576c 100%);
    border-radius: 15px;
    text-align: center;

    /* ТЕНЬ для цитаты */
    box-shadow: 0 8px 20px rgba(245, 87, 108, 0.2);

    color: white;
    position: relative;
    overflow: hidden;
}

.цитата::before {
    content: "“";
    position: absolute;
    top: 10px;
    left: 20px;
    font-size: 4rem;
    opacity: 0.2;
}

.цитата p {
    font-size: 1.3rem;
    font-style: italic;
    position: relative;
    z-index: 1;
}

/* ===== ГАЛЕРЕЯ РАЗНЫХ ТЕНЕЙ ===== */

.галерея-теней {
    background: white;
    padding: 30px;
    border-radius: 15px;
    margin: 40px auto;
```

```
max-width: 800px;

/* Рамка с тенью для самой галереи */
border: 1px solid #eee;
box-shadow: 0 5px 20px rgba(0,0,0,0.05);
}

.галерей-тений h3 {
    text-align: center;
    color: #2c3e50;
    margin-bottom: 25px;
}

.тень-варианты {
    display: grid;
    grid-template-columns: repeat(auto-fit, minmax(150px, 1fr));
    gap: 20px;
}

.вариант {
    text-align: center;
    cursor: pointer;
    padding: 15px;
    border-radius: 10px;
    transition: all 0.3s ease;
}

.вариант:hover {
    background-color: #f8f9fa;
    transform: translateY(-5px);
}

.предпросмотр-тени {
    width: 100px;
    height: 100px;
    background: white;
    border-radius: 10px;
    margin: 0 auto 15px;
    border: 1px solid #eee;
}

/* Разные варианты теней для предпросмотра */
#лёгкая-тень .предпросмотр-тени {
    box-shadow: 0 3px 10px rgba(0,0,0,0.08);
```

```
}

#выраженная-тень .предпросмотр-тени {
    box-shadow: 0 10px 25px rgba(0,0,0,0.15);
}

#цветная-тень .предпросмотр-тени {
    box-shadow: 0 5px 15px rgba(139, 69, 19, 0.2);
}

#двойная-тень .предпросмотр-тени {
    box-shadow:
        0 3px 10px rgba(0,0,0,0.1),
        0 10px 30px rgba(0,0,0,0.05);
}

/* ===== ПАНЕЛЬ УПРАВЛЕНИЯ ===== */
.панель-управления {
    background: white;
    padding: 30px;
    border-radius: 15px;
    margin: 40px auto;
    max-width: 800px;

    /* Красивая рамка с градиентом */
    border: 2px solid;
    border-image: linear-gradient(45deg, #3498db, #2ecc71) 1;

    /* Мягкая тень */
    box-shadow: 0 8px 25px rgba(0,0,0,0.1);
}

.панель-управления h3 {
    text-align: center;
    color: #2c3e50;
    margin-bottom: 25px;
    font-size: 1.5rem;
}

.управление-группа {
    margin-bottom: 20px;
}

.управление-группа label {
```

```
display: block;
margin-bottom: 8px;
color: #555;
font-weight: 600;
}

input[type="range"] {
    width: 100%;
    height: 10px;
    border-radius: 5px;
    background: #f0f0f0;
    outline: none;
}

.КНОПКИ {
    display: flex;
    justify-content: center;
    gap: 20px;
    margin-top: 30px;
}

.КНОПКА {
    padding: 12px 25px;
    background: #3498db;
    color: white;
    border: none;
    border-radius: 8px;
    font-size: 1rem;
    cursor: pointer;
    transition: all 0.3s ease;

    /* Тень для кнопки */
    box-shadow: 0 4px 10px rgba(52, 152, 219, 0.3);
}

.КНОПКА:hover {
    background: #2980b9;
    transform: translateY(-3px);
    box-shadow: 0 6px 15px rgba(52, 152, 219, 0.4);
}

#кнопка-сброс {
    background: #e74c3c;
    box-shadow: 0 4px 10px rgba(231, 76, 60, 0.3);
```

```
}

#кнопка-сброс:hover {
    background: #c0392b;
    box-shadow: 0 6px 15px rgba(231, 76, 60, 0.4);
}

/* ===== ПОДСКАЗКА ===== */
.подсказка {
    background: #fff8e1;
    padding: 20px;
    border-radius: 10px;
    margin-top: 30px;
    border-left: 5px solid #ffb300;
    color: #5d4037;
}

/* ===== АДАПТИВНОСТЬ ===== */
@media (max-width: 768px) {
    .контейнер {
        padding: 10px;
    }

    .заголовок-страницы {
        font-size: 2rem;
    }

    .визитка {
        padding: 25px;
        border-radius: 15px;
    }

    .информация {
        grid-template-columns: 1fr;
    }

    .тень-варианты {
        grid-template-columns: repeat(2, 1fr);
    }
}

@media (max-width: 480px) {
    .тень-варианты {
        grid-template-columns: 1fr;
```

```
}

.KНОПКИ {
    flex-direction: column;
}

}
```

---



## Шаг 5: Добавляем интерактивность (скрипт.js)

Создайте файл скрипт.js и добавьте следующий код:

javascript

```
// Ждём, пока вся страница загрузится
document.addEventListener('DOMContentLoaded', function() {
    // Получаем элементы управления
    const ползунокX = document.getElementById('ползунок-x');
    const ползунокY = document.getElementById('ползунок-y');
    const ползунокРазмытие = document.getElementById('ползунок-размытие');
    const ползунокПрозрачность = document.getElementById('ползунок-прозрачность');

    const значениеX = document.getElementById('значение-x');
    const значениеY = document.getElementById('значение-y');
    const значениеРазмытие = document.getElementById('значение-размытие');
    const значениеПрозрачность = document.getElementById('значение-прозрачность');

    const кнопкаСброс = document.getElementById('кнопка-сброс');
    const кнопкаКод = document.getElementById('кнопка-код');

    const визитка = document.getElementById('основная-визитка');

    // Варианты теней из галереи
    const вариантыТеней = {
        'лёгкая-тень': document.getElementById('лёгкая-тень'),
        'выраженная-тень': document.getElementById('выраженная-тень'),
        'цветная-тень': document.getElementById('цветная-тень'),
        'двойная-тень': document.getElementById('двойная-тень')
    };

    // Функция для обновления тени
    function обновитьТень() {
        const x = ползунокX.value;
```

```
const y = ползунокY.value;
const размытие = ползунокРазмытие.value;
const прозрачность = (ползунокПрозрачность.value / 100).toFixed(2);

// Обновляем отображаемые значения
значениеX.textContent = x;
значениеY.textContent = y;
значениеРазмытие.textContent = размытие;
значениеПрозрачность.textContent = прозрачность;

// Создаём CSS-значение для box-shadow
const тень = `${x}px ${y}px ${размытие}px rgba(0, 0, 0, ${прозрачность})`;

// Применяем тень к визитке
визитка.style.boxShadow = тень;
}

// Функция сброса к значениям по умолчанию
function сброситьТень() {
    ползунокX.value = 5;
    ползунокY.value = 5;
    ползунокРазмытие.value = 15;
    ползунокПрозрачность.value = 20;

    обновитьТень();

    // Показываем сообщение
    alert('Тень сброшена к значениям по умолчанию!');
}

// Функция показа CSS-кода
function показатьКод() {
    const x = ползунокX.value;
    const y = ползунокY.value;
    const размытие = ползунокРазмытие.value;
    const прозрачность = (ползунокПрозрачность.value / 100).toFixed(2);

    const cssКод = `.ваш-элемент {\n  box-shadow: ${x}px ${y}px ${размытие}px rgba(0, 0, 0, ${прозрачность});\n}`;

    alert('Скопируйте этот CSS-код:\n\n' + cssКод);
}

// Функция применения готового варианта тени
```

```

function применитьВариант(тип) {
    switch(тип) {
        case 'лёгкая-тень':
            ползунокX.value = 3;
            ползунокY.value = 3;
            ползунокРазмытие.value = 10;
            ползунокПрозрачность.value = 8;
            break;

        case 'выраженная-тень':
            ползунокX.value = 10;
            ползунокY.value = 10;
            ползунокРазмытие.value = 25;
            ползунокПрозрачность.value = 30;
            break;

        case 'цветная-тень':
            ползунокX.value = 5;
            ползунокY.value = 5;
            ползунокРазмытие.value = 15;
            ползунокПрозрачность.value = 20;
            // Для цветной тени немного сложнее - поменяем цвет вручную
            const прозрачность = 0.2;
            визитка.style.boxShadow = `5px 5px 15px rgba(139, 69, 19, ${прозрачность})`;
            return; // Выходим, чтобы не перезаписать тень

        case 'двойная-тень':
            const x = 3;
            const y = 3;
            const размытие1 = 10;
            const размытие2 = 30;
            const прозрачность1 = 0.1;
            const прозрачность2 = 0.05;

            визитка.style.boxShadow =
                `${x}px ${y}px ${размытие1}px rgba(0,0,0,${прозрачность1}), ` +
                `${x*2}px ${y*2}px ${размытие2}px rgba(0,0,0,${прозрачность2})`;
            return; // Выходим, чтобы не перезаписать тень
    }

    обновитьТень();
}

// Назначаем обработчики событий на ползунки

```

```
ползунокX.addEventListener('input', обновитьТень);
ползунокY.addEventListener('input', обновитьТень);
ползунокРазмытие.addEventListener('input', обновитьТень);
ползунокПрозрачность.addEventListener('input', обновитьТень);

// Назначаем обработчики на кнопки
кнопкаСброс.addEventListener('click', сброситьТень);
кнопкаКод.addEventListener('click', показатьКод);

// Назначаем обработчики на варианты теней
for (const [id, элемент] of Object.entries(вариантыТеней)) {
    элемент.addEventListener('click', function() {
        // Убираем выделение со всех вариантов
        for (const el of Object.values(вариантыТеней)) {
            el.style.backgroundColor = '';
            el.style.boxShadow = 'none';
        }

        // Выделяем выбранный вариант
        this.style.backgroundColor = '#f0f7ff';
        this.style.boxShadow = '0 3px 10px rgba(52, 152, 219, 0.2)';

        // Применяем выбранную тень
        применитьВариант(id);
    });
}

// Инициализируем тень при загрузке
обновитьТень();

// Добавляем интерактивность для информационных блоков
const инфоБлоки = document.querySelectorAll('.инфо-блок');
инфоБлоки.forEach(блок => {
    блок.addEventListener('mouseenter', function() {
        this.style.transform = 'translateY(-5px)';
        this.style.boxShadow =
            'inset 0 2px 5px rgba(0,0,0,0.05), ' +
            '0 8px 20px rgba(52, 152, 219, 0.15)';
    });

    блок.addEventListener('mouseleave', function() {
        this.style.transform = 'translateY(0)';
        this.style.boxShadow = 'inset 0 2px 5px rgba(0,0,0,0.05)';
    });
})
```

```
});  
  
// Интересный эффект для визитки при загрузке  
setTimeout(() => {  
    визитка.style.transform = 'scale(1.02)';  
    визитка.style.boxShadow = '0 20px 50px rgba(0, 0, 0, 0.15)';  
  
    setTimeout(() => {  
        визитка.style.transform = '';  
        обновитьТень();  
    }, 300);  
}, 500);  
});
```

---



## Шаг 6: Запускаем и экспериментируем!

1. Сохраните все три файла в одной папке Моя\_Визитка\_C\_Тенью
  2. Откройте файл index.html в браузере (двойной клик)
  3. Вы увидите:
    - Красивую визитку с лёгкой тенью
    - Галерею с разными вариантами теней
    - Панель управления с ползунками
  4. Экспериментируйте:
    - Наводите курсор на визитку — она поднимется
    - Нажмайте на варианты теней в галерее
    - Двигайте ползунки и наблюдайте изменения в реальном времени
    - Нажмите "Показать CSS-код" — получите готовый код для вашего проекта
- 



## Что мы изучили в этом практикуме:

### 1. Свойство box-shadow:

css

box-shadow: [X]px [Y]px [размытие]px [цвет];

## 2. Разные типы теней:

- **Лёгкие** (0 3px 10px rgba(0,0,0,0.08)) — для деликатных элементов
- **Выраженные** (0 10px 25px rgba(0,0,0,0.15)) — для акцентов
- **Цветные** (0 5px 15px rgba(139,69,19,0.2)) — для создания настроения
- **Двойные** — для сложных, многослойных эффектов

## 3. Внутренние тени (inset):

css

`box-shadow: inset 0 2px 5px rgba(0,0,0,0.05);`

## 4. Тени при наведении:

css

```
.element:hover {  
    box-shadow: 0 15px 35px rgba(0,0,0,0.12);  
    transform: translateY(-10px);  
}
```

## 5. Комбинирование с другими свойствами:

- `border-radius` — закруглённые углы + тень = современный вид
  - `border` — рамка + тень = глубина
  - `transition` — плавные анимации
- 

## 🏆 Домашнее задание:

### 1. Создайте визитку для друга/родственника

Используйте другой цвет тени, подходящий тематике (например, зелёную тень для садовода)

### 2. Сделайте «плавающую» фотогалерею

3-4 фотографии, каждая в рамке с тенью, при наведении поднимаются

### 3. Эксперимент с формами

Попробуйте разные значения `border-radius` с тенями:

- Круг (`border-radius: 50%`)
- Овал
- Скруглённые углы разной степени

#### 4. Создайте кнопку с 3D-эффектом

Используйте внутреннюю тень (inset) для эффекта нажатия

**Помните:** Лучший способ научиться — экспериментировать! Пробуйте странные значения, смотрите, что получается. CSS — это безопасная цифровая песочница. Даже если что-то «сломается», просто нажмите Ctrl+Z или верните исходный код.

---



#### Главный секрет красивых теней:

**Лучшая тень — та, которую почти не замечают.**

Она не кричит «Смотри на меня!», а тихо делает элемент объёмным, отделяет его от фона и создаёт ощущение качества и продуманности дизайна.

Поздравляю! Теперь вы умеете создавать профессионально выглядящие элементы с помощью рамок и теней. В следующей главе мы научимся управлять размером наших «рамок» — делать их шире, уже, выше или ниже. Это будет ещё один шаг к созданию полноценных, красивых веб-страниц!

## ● Глава 6: Как управлять размером «рамки»?

- Ширина (width) и высота (height): в пикселях и процентах.

Дорогой друг! Представьте, что вы делаете рамку для любимой фотографии. Вы можете выбрать точный размер: 10×15 см (это пиксели) или сказать «сделай рамку в половину размера стола» (это проценты). В CSS у нас есть те же возможности! Давайте научимся управлять шириной и высотой элементов.

---



### Что такое width и height?

Это два самых основных свойства CSS, которые контролируют размеры элемента:

- width — ширина элемента (от левого до правого края)
- height — высота элемента (от верхнего до нижнего края)

css

```
.моя-рамка {  
    width: 300px; /* Ширина 300 пикселей */  
    height: 200px; /* Высота 200 пикселей */  
}
```

---

### 🎯 1. Абсолютные единицы: Пиксели (px) — «Точная линейка»

**Пиксели** — это как сантиметры в реальной жизни. Точные, конкретные значения.

css

```
.точный-размер {  
    width: 300px; /* Всегда 300 пикселей */  
    height: 150px; /* Всегда 150 пикселей */  
}
```

## Примеры использования px:

### A) Фотографии фиксированного размера:

css

```
.фотография {  
    width: 800px;  
    height: 600px;  
    border: 5px solid #ccc;  
    box-shadow: 0 5px 15px rgba(0,0,0,0.1);  
}
```

### Б) Кнопки с точными размерами:

css

```
.кнопка {  
    width: 120px;  
    height: 40px;  
    background: #3498db;  
    color: white;  
    border: none;  
    border-radius: 5px;  
}
```

### В) Элементы интерфейса:

css

```
.иконка {  
    width: 24px; /* Стандартный размер иконок */  
    height: 24px;  
}
```

## 🔍 Практический пример: Галерея фотографий

Создайте файл галерея-пиксели.html:

html

```
<!DOCTYPE html>  
<html lang="ru">  
<head>  
    <meta charset="UTF-8">  
    <meta name="viewport" content="width=device-width, initial-scale=1.0">  
    <title>Галерея с фиксированными размерами</title>
```

```
<style>

body {
    font-family: 'Arial', sans-serif;
    background: #f5f5f5;
    padding: 20px;
}

.галерея {
    display: flex;
    flex-wrap: wrap;
    gap: 20px;
    justify-content: center;
}

.фото-рамка {
    /* ФИКСИРОВАННЫЕ РАЗМЕРЫ в пикселях */
    width: 250px;
    height: 200px;

    background: white;
    border-radius: 10px;
    overflow: hidden;
    box-shadow: 0 5px 15px rgba(0,0,0,0.1);
    transition: transform 0.3s;
}

.фото-рамка:hover {
    transform: scale(1.05);
    box-shadow: 0 10px 25px rgba(0,0,0,0.15);
}

.фото {
    width: 100%; /* Занимает всю ширину рамки */
    height: 140px; /* Фиксированная высота фото */
    background: linear-gradient(45deg, #667eea, #764ba2);
    display: flex;
    align-items: center;
    justify-content: center;
    color: white;
    font-size: 18px;
}

.описание {
    padding: 15px;
```

```
        text-align: center;
        color: #333;
    }

    .контрольная-панель {
        background: white;
        padding: 20px;
        margin: 30px auto;
        max-width: 500px;
        border-radius: 10px;
        box-shadow: 0 5px 15px rgba(0,0,0,0.1);
    }

    .ползунок {
        width: 100%;
        margin: 10px 0;
    }

```

</style>

</head>

<body>

<h1 style="text-align: center;">Галерея с фиксированными размерами (px)</h1>

<div class="галерея">

<div class="фото-рамка" id="фото1">

<div class="фото"> Фото 1</div>

<div class="описание">Розы в саду<br>250×200px</div>

</div>

<div class="фото-рамка" id="фото2">

<div class="фото"> Фото 2</div>

<div class="описание">Подсолнухи<br>250×200px</div>

</div>

<div class="фото-рамка" id="фото3">

<div class="фото"> Фото 3</div>

<div class="описание">Сакура цветёт<br>250×200px</div>

</div>

<div class="фото-рамка" id="фото4">

<div class="фото"> Фото 4</div>

<div class="описание">Тюльпаны весной<br>250×200px</div>

</div>

</div>

```
<div class="контрольная-панель">
    <h3>Эксперимент: Изменяйте размеры в пикселях</h3>

    <label>Ширина рамок: <span id="ширина-значение">250</span>px</label>
    <input type="range" class="ползунок" id="ширина-ползунок" min="100" max="400"
value="250">

    <label>Высота рамок: <span id="высота-значение">200</span>px</label>
    <input type="range" class="ползунок" id="высота-ползунок" min="100" max="400"
value="200">

    <div style="margin-top: 20px;">
        <button onclick="сброситьРазмеры()">Сбросить</button>
        <button onclick="показатьКод()">Показать CSS-код</button>
    </div>
</div>

<script>
    const ползунокШирина = document.getElementById('ширина-ползунок');
    const ползунокВысота = document.getElementById('высота-ползунок');
    const значениеШирина = document.getElementById('ширина-значение');
    const значениеВысота = document.getElementById('высота-значение');
    const рамки = document.querySelectorAll('.фото-рамка');

    function обновитьРазмеры() {
        const ширина = ползунокШирина.value;
        const высота = ползунокВысота.value;

        значениеШирина.textContent = ширина;
        значениеВысота.textContent = высота;

        рамки.forEach(рамка => {
            рамка.style.width = ширина + 'px';
            рамка.style.height = высота + 'px';
        });
    }

    function сброситьРазмеры() {
        ползунокШирина.value = 250;
        ползунокВысота.value = 200;
        обновитьРазмеры();
    }

    function показатьКод() {
```

```

const ширина = ползунокШирина.value;
const высота = ползунокВысота.value;
alert(`CSS-код:\n\n.фото-рамка {\n  width: ${ширина}px;\n  height: ${высота}px;\n}`);
}

ползунокШирина.addEventListener('input', обновитьРазмеры);
ползунокВысота.addEventListener('input', обновитьРазмеры);

// Инициализация
обновитьРазмеры();
</script>
</body>
</html>
```

### Преимущества px:

- **Точность** — всегда один и тот же размер
- **Предсказуемость** — знаете точно, что получите
- **Простота** — легко использовать и понимать

### Недостатки px:

- **Не адаптивно** — на маленьком экране 300px могут занимать пол-экрана
  - **Жёсткость** — не подстраивается под размер экрана
  - **Неудобно для пользователей** — кто-то хочет увеличить текст, но элементы не масштабируются
- 

## 2. Относительные единицы: Проценты (%) — «Резиновые» размеры

**Проценты** — это относительные единицы. Они зависят от размера родительского элемента.

css

```
.относительный-размер {
  width: 50%; /* Половина ширины родителя */
  height: 75%; /* Три четверти высоты родителя */
}
```

## Как это работает:

```
html

<div class="родитель" style="width: 800px;">
    <div class="ребёнок" style="width: 50%;">
        <!-- Этот элемент будет шириной 400px (50% от 800px) --&gt;
    &lt;/div&gt;
&lt;/div&gt;</pre>
```

## 🔍 Практический пример: Адаптивная галерея

Создайте файл галерея-проценты.html:

```
html

<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Адаптивная галерея с процентами</title>
    <style>
        * {
            box-sizing: border-box;
        }

        body {
            font-family: 'Arial', sans-serif;
            background: #f5f5f5;
            margin: 0;
            padding: 20px;
        }

        .контейнер {
            /* Контейнер занимает 90% ширины окна */
            width: 90%;
            max-width: 1200px;
            margin: 0 auto;
            background: white;
            padding: 30px;
            border-radius: 15px;
            box-shadow: 0 5px 20px rgba(0,0,0,0.1);
        }
    </style>
</head>
<body>
    <div class="контейнер">
        <!-- Содержимое галереи -->
    </div>
</body>
</html>
```

```
.галерея {  
    display: flex;  
    flex-wrap: wrap;  
    gap: 2%; /* Процентные отступы между элементами */  
    margin: 30px 0;  
}  
  
.фото-карточка {  
    /* КАЖДАЯ КАРТОЧКА занимает процент ширины контейнера */  
    width: 23%; /* 4 карточки в ряд с отступами */  
  
    /* Высота пропорциональна ширине */  
    height: 0;  
    padding-bottom: 23%; /* Создаёт квадрат */  
  
    background: linear-gradient(45deg, #667eea, #764ba2);  
    border-radius: 10px;  
    position: relative;  
    overflow: hidden;  
    transition: all 0.3s ease;  
    box-shadow: 0 5px 15px rgba(0,0,0,0.1);  
}  
  
.фото-карточка:hover {  
    transform: scale(1.05);  
    box-shadow: 0 10px 25px rgba(0,0,0,0.2);  
}  
  
.содержимое {  
    position: absolute;  
    top: 0;  
    left: 0;  
    width: 100%;  
    height: 100%;  
    display: flex;  
    flex-direction: column;  
    justify-content: center;  
    align-items: center;  
    color: white;  
    padding: 20px;  
    text-align: center;  
}  
  
.номер {
```

```
        font-size: 2rem;
        font-weight: bold;
        margin-bottom: 10px;
    }

.описание {
    font-size: 0.9rem;
    opacity: 0.9;
}

/* Инфо-панель */
.инфо-панель {
    background: #f8f9fa;
    padding: 20px;
    border-radius: 10px;
    margin: 20px 0;
    border-left: 5px solid #3498db;
}

/* Управление */
.управление {
    background: #2c3e50;
    color: white;
    padding: 25px;
    border-radius: 10px;
    margin-top: 30px;
}

.ползунок {
    width: 100%;
    margin: 10px 0;
}

/* Адаптивность */
@media (max-width: 900px) {
    .фото-карточка {
        width: 31.33%; /* 3 карточки в ряд */
        padding-bottom: 31.33%;
    }
}

@media (max-width: 600px) {
    .фото-карточка {
        width: 48%; /* 2 карточки в ряд */
    }
}
```

```

        padding-bottom: 48%;
    }
}

@media (max-width: 400px) {
    .фото-карточка {
        width: 100%; /* 1 карточка в ряд */
        padding-bottom: 100%;
    }
}

</style>
</head>
<body>

    <div class="контейнер">
        <h1 style="text-align: center; color: #2c3e50;">Адаптивная галерея с процентами</h1>
        <p style="text-align: center; color: #7f8c8d;">Измените размер окна браузера и посмотрите,  

как карточки автоматически подстраиваются!</p>

        <div class="инфо-панель">
            <h3>⌚ Как работают проценты?</h3>
            <p>Каждая карточка имеет <code>width: 23%</code>. Это значит:</p>
            <ul>
                <li>На широком экране: 4 карточки в ряд ( $23\% \times 4 + \text{отступы}$ )</li>
                <li>На планшете: 3 карточки в ряд</li>
                <li>На телефоне: 2 или 1 карточка в ряд</li>
            </ul>
            <p><strong>Проценты всегда считаются от ширины родительского элемента!</strong></p>
        </div>

        <div class="галерея" id="галерея">
            <!-- Карточки будут созданы скриптом --&gt;
        &lt;/div&gt;

        &lt;div class="управление"&gt;
            &lt;h3 style="margin-top: 0;"&gt;🎨 Лаборатория процентов&lt;/h3&gt;

            &lt;label&gt;Ширина карточки: &lt;span id="ширина-значение"&gt;23&lt;/span&gt;%&lt;/label&gt;
            &lt;input type="range" class="ползунок" id="ширина-ползунок" min="10" max="100"
value="23"&gt;

            &lt;label&gt;Отступы между карточками: &lt;span id="отступ-значение"&gt;2&lt;/span&gt;%&lt;/label&gt;
            &lt;input type="range" class="ползунок" id="отступ-ползунок" min="0" max="10"
value="2"&gt;
        </pre>

```

```
<label>Количество карточек: <span id="количество-значение">12</span></label>
<input type="range" class="ползунок" id="количество-ползунок" min="4" max="20"
value="12">

<div style="margin-top: 20px;">
    <button onclick="создатьКарточки()" style="background: #3498db; color: white;
border: none; padding: 10px 20px; border-radius: 5px; cursor: pointer;">
        Обновить галерею
    </button>
    <button onclick="сброситьНастройки()" style="background: #e74c3c; color: white;
border: none; padding: 10px 20px; border-radius: 5px; cursor: pointer;">
        Сбросить
    </button>
</div>
</div>

<div style="background: #e8f4fc; padding: 20px; border-radius: 10px; margin-top: 30px;
border-left: 5px solid #3498db;">
    <h3>💡 Практический пример: Блок с процентной шириной</h3>

    <div style="background: white; padding: 20px; border-radius: 8px; margin: 15px 0;
border: 2px dashed #3498db;">
        <div style="width: 80%; height: 40px; background: #3498db; margin: 10px auto;
border-radius: 5px;"></div>
        <p style="text-align: center;">Этот блок имеет <code>width: 80%</code> — 80% от
ширины родительского контейнера</p>
    </div>

    <div style="width: 100%; background: #f0f0f0; height: 200px; position: relative; border-
radius: 8px; overflow: hidden;">
        <div id="пример-блок" style="position: absolute; top: 20px; left: 10%; width: 50%;
height: 80%; background: #2ecc71; border-radius: 5px; display: flex; align-items: center; justify-content:
center; color: white; font-weight: bold;">
            50% ширины
        </div>
        <div style="position: absolute; top: 20px; right: 10%; width: 30%; height: 80%;
background: #e74c3c; border-radius: 5px; display: flex; align-items: center; justify-content: center; color:
white; font-weight: bold;">
            30% ширины
        </div>
    </div>
</div>
```

<p style="margin-top: 10px;"><strong>Попробуйте:</strong> Измените ширину окна браузера и посмотрите, как блоки с процентами меняют размер!</p>

```
</div>  
</div>
```

```
<script>  
    // Цвета для карточек  
    const цвета = [  
        'linear-gradient(45deg, #667eea, #764ba2)',  
        'linear-gradient(45deg, #f093fb, #f5576c)',  
        'linear-gradient(45deg, #4facfe, #00f2fe)',  
        'linear-gradient(45deg, #43e97b, #38f9d7)',  
        'linear-gradient(45deg, #fa709a, #fee140)',  
        'linear-gradient(45deg, #30cf0, #330867)'  
    ];
```

```
    // Названия для карточек  
    const названия = [  
        'Весенний сад', 'Летний букет', 'Осенние листья',  
        'Зимние узоры', 'Цветущий луг', 'Розовый рассвет',  
        'Морская волна', 'Золотой закат', 'Лесная тропа',  
        'Горный пейзаж', 'Деревенский дворик', 'Парковая аллея'  
    ];
```

```
    // Создаём карточки  
    function создатьКарточки() {  
        const галерея = document.getElementById('галерея');  
        const количество = document.getElementById('количество-ползунок').value;  
  
        // Очищаем галерею  
        галерея.innerHTML = "";  
  
        // Создаём новые карточки  
        for (let i = 0; i < количество; i++) {  
            const карточка = document.createElement('div');  
            карточка.className = 'фото-карточка';  
            карточка.style.background = цвета[i % цвета.length];  
  
            const содержимое = document.createElement('div');  
            содержимое.className = 'содержимое';  
  
            const номер = document.createElement('div');  
            номер.className = 'номер';  
            номер.textContent = `#${i + 1}`;  
        }  
    }
```

```

const описание = document.createElement('div');
описание.className = 'описание';
описание.textContent = названия[i % названия.length];

содержимое.appendChild(номер);
содержимое.appendChild(описание);
карточка.appendChild(содержимое);
галерея.appendChild(карточка);
}

обновитьРазмеры();
}

// Обновляем размеры карточек
function обновитьРазмеры() {
    const ширина = document.getElementById('ширина-ползунок').value;
    const отступ = document.getElementById('отступ-ползунок').value;
    const количество = document.getElementById('количество-значение');

    // Обновляем отображаемые значения
    document.getElementById('ширина-значение').textContent = ширина;
    document.getElementById('отступ-значение').textContent = отступ;
    количество.textContent = document.getElementById('количество-ползунок').value;

    // Обновляем CSS переменные
    const галерея = document.getElementById('галерея');
    const карточки = галерея.querySelectorAll('.фото-карточка');

    // Применяем новые размеры
    карточки.forEach(карточка => {
        карточка.style.width = ширина + '%';
        карточка.style.paddingBottom = ширина + '%';
    });

    // Обновляем отступы
    галерея.style.gap = отступ + '%';
}

// Сброс настроек
function сброситьНастройки() {
    document.getElementById('ширина-ползунок').value = 23;
    document.getElementById('отступ-ползунок').value = 2;
    document.getElementById('количество-ползунок').value = 12;
}

```

```

        обновитьРазмеры();
        создатьКарточки();
    }

    // Назначаем обработчики
    document.getElementById('ширина-ползунок').addEventListener('input', обновитьРазмеры);
    document.getElementById('отступ-ползунок').addEventListener('input', обновитьРазмеры);
    document.getElementById('количество-ползунок').addEventListener('input', function() {
        document.getElementById('количество-значение').textContent = this.value;
   });

    // Инициализация
    создатьКарточки();

    // Пример для демонстрации
    const примерБлок = document.getElementById('пример-блок');
    let процент = 50;
    let направление = 1;

    setInterval(() => {
        процент += направление;
        if (процент >= 80 || процент <= 20) направление *= -1;
        примерБлок.style.width = процент + '%';
        примерБлок.textContent = процент + '% ширины';
    }, 100);

```

</script>

</body>

</html>

### Преимущества %:

- **Адаптивность** — автоматически подстраивается под размер экрана
- **Гибкость** — легко создавать резиновые макеты
- **Относительность** — размеры зависят от контекста
- **Простота создания колонок** — 50% = две колонки, 33.33% = три колонки

### Недостатки %:

- **Непредсказуемость** — зависит от родителя
  - **Сложность вложенности** — проценты от процентов могут дать неожиданный результат
  - **Нужно знать контекст** — важно понимать, от чего считаются проценты
-

## 3. Когда что использовать?

**Используйте пиксели (px), когда:**

1. **Нужна точность** — иконки, маленькие элементы интерфейса
2. **Фиксированный макет** — сайт должен выглядеть одинаково на всех экранах
3. **Границы и тени** — border: 1px solid #ccc;
4. **Изображения фиксированного размера** — когда знаете точные размеры

**Используйте проценты (%), когда:**

1. **Адаптивный дизайн** — сайт должен подстраиваться под разные экраны
  2. **Создание колонок** — сетки, галереи, карточки товаров
  3. **Ширина контейнеров** — width: 90%; max-width: 1200px;
  4. **Высота относительно родителя** — height: 100%;
- 

## 4. Комбинирование px и %

Часто лучший подход — комбинировать обе единицы:

css

```
.контейнер {  
    width: 90%;      /* 90% от окна браузера */  
    max-width: 1200px; /* Но не больше 1200px */  
    min-width: 300px; /* И не меньше 300px */  
    margin: 0 auto;   /* Центрирование */  
}  
  
.блок {  
    width: 300px;    /* Фиксированная ширина */  
    height: 50%;     /* Половина высоты родителя */  
}  
  
.кнопка {  
    width: 120px;    /* Фиксированная ширина */  
    height: 40px;    /* Фиксированная высота */  
    font-size: 16px;  /* Фиксированный размер шрифта */  
}
```

---

## 🎓 5. Практикум: Создаём адаптивную карточку товара

Создайте файл `карточка-товара.html`:

html

```
<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Адаптивная карточка товара</title>
    <style>
        * {
            box-sizing: border-box;
        }

        body {
            font-family: 'Arial', sans-serif;
            background: #f5f5f5;
            padding: 20px;
            display: flex;
            justify-content: center;
            align-items: center;
            min-height: 100vh;
        }

        .карточка {
            /* Комбинируем px и % */
            width: 90%;      /* 90% от родителя (body) */
            max-width: 400px; /* Но не больше 400px */
            min-width: 300px; /* И не меньше 300px */

            background: white;
            border-radius: 15px;
            overflow: hidden;
            box-shadow: 0 10px 30px rgba(0,0,0,0.1);
            transition: transform 0.3s;
        }

        .карточка:hover {
            transform: translateY(-10px);
            box-shadow: 0 15px 40px rgba(0,0,0,0.15);
        }
    </style>
</head>
<body>
    <div class="карточка">
        Текст о товаре
    </div>
</body>
</html>
```

```
.изображение {
    /* Высота в пикселях, ширина в процентах */
    width: 100%;      /* Вся ширина карточки */
    height: 250px;    /* Фиксированная высота */
    background: linear-gradient(45deg, #667eea, #764ba2);
    display: flex;
    align-items: center;
    justify-content: center;
    color: white;
    font-size: 2rem;
}

.контент {
    padding: 25px;
}

.заголовок {
    color: #2c3e50;
    margin-bottom: 10px;
    font-size: 1.5rem;
}

.описание {
    color: #7f8c8d;
    line-height: 1.6;
    margin-bottom: 20px;
}

.цена-блок {
    display: flex;
    justify-content: space-between;
    align-items: center;
    margin-bottom: 20px;
    padding-bottom: 20px;
    border-bottom: 1px solid #eee;
}

.цена {
    font-size: 1.8rem;
    color: #27ae60;
    font-weight: bold;
}
```

```
.рейтинг {  
    color: #f39c12;  
    font-size: 1.2rem;  
}  
  
.кнопки {  
    display: flex;  
    gap: 10px;  
}  
  
.кнопка {  
    flex: 1;      /* Равномерно распределяют пространство */  
    padding: 12px;  
    border: none;  
    border-radius: 8px;  
    font-size: 16px;  
    cursor: pointer;  
    transition: all 0.3s;  
}  
  
.кнопка.купить {  
    background: #3498db;  
    color: white;  
}  
  
.кнопка.в-корзину {  
    background: #f8f9fa;  
    color: #333;  
    border: 1px solid #ddd;  
}  
  
.кнопка:hover {  
    transform: translateY(-2px);  
    box-shadow: 0 5px 15px rgba(0,0,0,0.1);  
}  
  
/* Адаптивность */  
@media (max-width: 500px) {  
    .кнопки {  
        flex-direction: column;  
    }  
  
.изображение {  
    height: 200px; /* Меньшая высота на мобильных */  
}
```

```
        }

    }

    @media (max-width: 350px) {
        .карточка {
            min-width: 250px;
        }

        .изображение {
            height: 150px;
            font-size: 1.5rem;
        }

        .контент {
            padding: 15px;
        }
    }

/* Панель управления */
.управление {
    background: white;
    padding: 25px;
    border-radius: 15px;
    margin-top: 40px;
    max-width: 500px;
    box-shadow: 0 5px 20px rgba(0,0,0,0.1);
}

.ползунок {
    width: 100%;
    margin: 10px 0;
}

</style>
</head>
<body>
    <div style="max-width: 500px;">
        <div class="карточка" id="товар">
            <div class="изображение" id="изображение-товара">
                 Подарочный набор
            </div>

            <div class="контент">
                <h2 class="заголовок" id="заголовок-товара">Набор для садовода «Весеннее
настроение»</h2>
            
```

```
<p class="описание" id="описание-товара">  
    В набор входят: семена цветов, садовые перчатки, набор инструментов  
    и книга по садоводству. Идеальный подарок для любителей растений.  
</p>  
  
<div class="цена-блок">  
    <div class="цена" id="цена-товара">1 499 ₽</div>  
    <div class="рейтинг">★★★★☆ 4.2</div>  
</div>  
  
<div class="кнопки">  
    <button class="кнопка купить" id="кнопка-купить">Купить сейчас</button>  
    <button class="кнопка в-корзину" id="кнопка-корзина">В корзину</button>  
</div>  
</div>  
  
<div class="управление">  
    <h3 style="text-align: center;">🕒 Лаборатория размеров</h3>  
    <p style="text-align: center; color: #666;">Экспериментируйте с px и %</p>  
  
<div style="display: grid; grid-template-columns: 1fr 1fr; gap: 20px; margin-top: 20px;">  
    <div>  
        <label>Ширина карточки:</label>  
        <select id="ширина-тип" style="width: 100%; padding: 8px; margin-bottom: 10px;">  
            <option value="px">Пиксели (px)</option>  
            <option value="percent">Проценты (%)</option>  
        </select>  
        <input type="range" class="ползунок" id="ширина-значение" min="200" max="800" value="400">  
        <div style="text-align: center;" id="ширина-текст">400px</div>  
    </div>  
  
    <div>  
        <label>Высота изображения:</label>  
        <select id="высота-тип" style="width: 100%; padding: 8px; margin-bottom: 10px;">  
            <option value="px">Пиксели (px)</option>  
            <option value="percent">Проценты (%)</option>  
        </select>  
        <input type="range" class="ползунок" id="высота-значение" min="100" max="400" value="250">  
        <div style="text-align: center;" id="высота-текст">250px</div>  
    </div>  
</div>
```

```

        </div>
    </div>

    <div style="margin-top: 20px; padding: 15px; background: #f8f9fa; border-radius: 8px;">
        <h4>Текущий CSS-код:</h4>
        <code id="css-код">.карточка { width: 400px; }<br>.изображение { height:
250px; }</code>
    </div>

    <div style="margin-top: 20px; text-align: center;">
        <button onclick="сбросить()" style="background: #e74c3c; color: white; border:
none; padding: 10px 20px; border-radius: 5px; cursor: pointer;">
            Сбросить
        </button>
        <button onclick="изменитьРазмерОкна()" style="background: #3498db; color:
white; border: none; padding: 10px 20px; border-radius: 5px; cursor: pointer; margin-left: 10px;">
            Изменить размер окна
        </button>
    </div>
</div>

<script>
    const карточка = document.getElementById('товар');
    const изображение = document.getElementById('изображение-товара');
    const ширинаТип = document.getElementById('ширина-тип');
    const ширинаЗначение = document.getElementById('ширина-значение');
    const ширинаТекст = document.getElementById('ширина-текст');
    const высотаТип = document.getElementById('высота-тип');
    const высотаЗначение = document.getElementById('высота-значение');
    const высотаТекст = document.getElementById('высота-текст');
    const cssКод = document.getElementById('css-код');

    function обновитьРазмеры() {
        const ширинаТипVal = ширинаТип.value;
        const ширинаVal = ширинаЗначение.value;
        const высотаТипVal = высотаТип.value;
        const высотаVal = высотаЗначение.value;

        // Обновляем отображение
        ширинаТекст.textContent = ширинаVal + (ширинаТипVal === 'px' ? 'px' : '%');
        высотаТекст.textContent = высотаVal + (высотаТипVal === 'px' ? 'px' : '%');

        // Применяем стили
    }
</script>

```

```

карточка.style.width = ширинаVal + (ширинаТипVal === 'px' ? 'px' : '%');
изображение.style.height = высотаVal + (высотаТипVal === 'px' ? 'px' : '%');

// Обновляем CSS-код
cssКод.textContent = `.карточка { width: ${ширинаVal}${ширинаТипVal === 'px' ? 'px' :
'%'}; }` + `п.изображение { height: ${высотаVal}${высотаТипVal === 'px' ? 'px' : '%'}; }`;

}

function сбросить() {
    ширинаТип.value = 'px';
    ширинаЗначение.value = 400;
    высотаТип.value = 'px';
    высотаЗначение.value = 250;
    обновитьРазмеры();
}

function изменитьРазмерОкна() {
    // Меняем размер окна для демонстрации адаптивности
    window.resizeTo(500, 800);
    setTimeout(() => {
        window.resizeTo(800, 600);
    }, 2000);
}

// Назначаем обработчики
ширинаТип.addEventListener('change', обновитьРазмеры);
ширинаЗначение.addEventListener('input', обновитьРазмеры);
высотаТип.addEventListener('change', обновитьРазмеры);
высотаЗначение.addEventListener('input', обновитьРазмеры);

// Обработчики для кнопок товара
document.getElementById('кнопка-купить').addEventListener('click', function() {
    alert('Спасибо за покупку! Товар добавлен в заказ.');
});

document.getElementById('кнопка-корзина').addEventListener('click', function() {
    this.textContent = '✓ В корзине';
    this.style.background = '#2ecc71';
    this.style.color = 'white';
    setTimeout(() => {
        this.textContent = 'В корзину';
        this.style.background = '';
        this.style.color = '';
    }, 2000);
});

```

```
});  
  
    // Инициализация  
    обновитьРазмеры();  
  </script>  
</body>  
</html>
```

---



## Сравнительная таблица: px vs %

Критерий	Пиксели (px)	Проценты (%)
<b>Точность</b>	Абсолютная, фиксированная	Относительная, зависит от родителя
<b>Адаптивность</b>	Низкая (фиксированный размер)	Высокая (автоматически подстраивается)
<b>Использование</b>	Границы, тени, иконки, точные размеры	Ширина блоков, колонки, адаптивные макеты
<b>Предсказуемость</b>	Всегда одинаковый результат	Зависит от размера родительского элемента
<b>Пример</b>	width: 300px; (всегда 300px)	width: 50%; (половина от родителя)
<b>Лучше для</b>	Деталей, мелких элементов, фиксированных частей	Макета, структуры, больших блоков



## Итоги и домашнее задание

### Что мы узнали:

1. width и height — основные свойства для управления размерами
2. Пиксели (px) — абсолютные, фиксированные размеры
3. Проценты (%) — относительные размеры, зависящие от родителя
4. Когда что использовать — комбинируйте оба подхода для лучших результатов

## **Домашнее задание:**

### **Задание 1: «Фоторамка с выбором размера»**

Создайте страницу с фоторамкой, где можно выбрать:

- Фиксированный размер (300×400px)
- Относительный размер (50% ширины экрана)
- Кнопку переключения между px и %

### **Задание 2: «Адаптивная сетка фотографий»**

Создайте галерею из 6 фотографий, которая показывает:

- На компьютере: 3 фотографии в ряд
- На планшете: 2 фотографии в ряд
- На телефоне: 1 фотографию в ряд

Используйте проценты и медиа-запросы.

### **Задание 3: «Личный кабинет с колонками»**

Создайте макет личного кабинета с:

- Левой колонкой: 25% ширины (меню)
- Правой колонкой: 75% ширины (контент)
- Футер: 100% ширины

### **Задание 4: «Эксперимент сложенными процентами»**

Создайте 3 вложенных блока, каждый с width: 50%. Измерьте итоговую ширину самого внутреннего блока и объясните результат.

---



## **Золотые правила:**

1. **Начинайте с процентов** для больших блоков и структуры
2. **Используйте пиксели** для мелких деталей и точного контроля
3. **Комбинируйте** — часто лучший результат даёт width: 90%; max-width: 1200px;
4. **Тестируйте на разных экранах** — меняйте размер окна браузера во время разработки
5. **Не бойтесь экспериментировать** — пробуйте разные значения и смотрите, что получается

**Запомните:** Управление размерами — это как шитьё одежды. Иногда нужен точный размер (как у костюма), иногда — свободный покрой (как у халата). Выбирайте то, что подходит для каждой конкретной задачи!

Теперь вы умеете контролировать размеры элементов двумя основными способами. В следующей главе мы познакомимся с важным понятием `max-width`, которое не даст нашей «мебели» застрять в «дверном проёме» маленького экрана!

## ■ Важное понятие: **max-width** – чтобы наша «мебель» влезала в любой «дверной проем» (размер экрана).

Дорогой друг! Представьте, что вы купили новый шкаф. Он прекрасно встаёт в вашу просторную гостиную. Но что, если вам нужно перевезти его в меньшую комнату? Или показать другу на маленьком экране телефона? Вот для этого и существует волшебное свойство **max-width**! Оно говорит элементу: «Расти, но не больше указанного предела».

---

### 🎯 Что такое **max-width**?

**max-width** — это свойство CSS, которое устанавливает **максимальную ширину** элемента. Элемент может быть меньше этого значения, но никогда — больше.

css

```
.мой-блок {  
    max-width: 800px; /* Не может быть шире 800px */  
    width: 100%;   /* Но может занимать всю доступную ширину */  
}
```

### Простая аналогия:

- **width: 800px** = «Дверь должна быть ровно 80 см шириной»
  - **max-width: 800px** = «Дверь может быть любой ширины, но не шире 80 см»
- 

### 💡 Почему это важно? Метафора с мебелью

Давайте представим нашу веб-страницу как комнату:

Элемент	Аналогия в комнате	Проблема без <b>max-width</b>	Решение с <b>max-width</b>
Текст статьи	Диван	На широком экране — растягивается, строки	<b>max-width: 800px</b> — ограничиваем длину строк для удобного чтения

Элемент	Аналогия в комнате	Проблема без max-width	Решение с max-width
		становятся слишком длинными, читать неудобно	
Изображение	Картина	На телефоне выходит за края экрана, появляется горизонтальная прокрутка	max-width: 100% — картина уменьшается, чтобы поместиться
Контейнер сайта	Ковёр	На 4К-мониторе занимает всю ширину (2 метра!), выглядит пусто	max-width: 1200px — ковёр никогда не шире 1.2 метра
Боковая панель	Тумбочка	На узком экране вытесняет основной контент	max-width: 300px — тумбочка не разрастается



## Основы использования max-width

### 1. Для текстового контента (статьи, блоги)

css

```
.статья {
    max-width: 800px; /* Оптимальная ширина для чтения */
    margin: 0 auto; /* Центрирование */
    padding: 20px;
    line-height: 1.6;
}
```

**Почему 600-800px?** Психологически комфортная длина строки — 50-75 символов. Более длинные строки утомляют глаза.

### 2. Для изображений (адаптивность!)

css

```
img {
    max-width: 100%; /* Важнейшее правило! */
    height: auto; /* Сохраняем пропорции */
```

```
}
```

Это волшебная строчка делает все изображения адаптивными! Они никогда не выйдут за границы родительского элемента.

### 3. Для основного контейнера сайта

css

```
.container {  
    max-width: 1200px; /* Популярный стандарт */  
    width: 90%;      /* На узких экранах займёт 90% */  
    margin: 0 auto;   /* Центрирование */  
    padding: 0 15px;  
}
```

### 4. Комбинирование с min-width

css

```
.блок {  
    min-width: 300px; /* Не может быть уже 300px */  
    max-width: 800px; /* Не может быть шире 800px */  
    width: 100%;     /* Занимает всю доступную ширину в этих пределах */  
}
```

---

## Практический пример: Адаптивный контейнер

Создайте файл max-width-пример.html:

html

```
<!DOCTYPE html>  
<html lang="ru">  
<head>  
    <meta charset="UTF-8">  
    <meta name="viewport" content="width=device-width, initial-scale=1.0">  
    <title>Волшебство max-width: «умная» мебель для сайта</title>  
<style>  
    * {  
        box-sizing: border-box;  
        margin: 0;
```

```
padding: 0;
}

body {
    font-family: 'Georgia', serif;
    line-height: 1.6;
    color: #333;
    background: linear-gradient(135deg, #f5f7fa 0%, #c3cfe2 100%);
    min-height: 100vh;
    padding: 20px;
}

/* ===== БЕЗ MAX-WIDTH ===== */
.проблема .контейнер {
    width: 1200px; /* ЖЁСТКАЯ ШИРИНА! */
    margin: 0 auto;
    background: #ffebee;
    padding: 30px;
    border-radius: 10px;
    border: 3px solid #ef9a9a;
    margin-bottom: 40px;
}

/* ===== С MAX-WIDTH ===== */
.решение .контейнер {
    max-width: 1200px; /* МАКСИМУМ 1200px */
    width: 90%; /* На узких экранах - 90% */
    margin: 0 auto; /* Центрирование */
    background: #e8f5e9;
    padding: 30px;
    border-radius: 10px;
    border: 3px solid #81c784;
}

/* Общие стили */
h1, h2 {
    text-align: center;
    margin-bottom: 20px;
    color: #2c3e50;
}

h1 {
    margin-bottom: 40px;
    padding-bottom: 15px;
```

```
border-bottom: 3px solid #3498db;
}

.раздел {
    margin-bottom: 50px;
}

.метафора {
    background: white;
    padding: 25px;
    border-radius: 15px;
    margin: 25px 0;
    box-shadow: 0 5px 20px rgba(0,0,0,0.1);
    border-left: 8px solid;
}

.плохая-метафора {
    border-left-color: #e74c3c;
    background: #ffebee;
}

.хорошая-метафора {
    border-left-color: #2ecc71;
    background: #e8f5e9;
}

/* Пример с изображениями */
.пример-изображений {
    display: flex;
    gap: 20px;
    margin: 30px 0;
    flex-wrap: wrap;
}

.изображение-блок {
    flex: 1;
    min-width: 300px;
    padding: 20px;
    background: white;
    border-radius: 10px;
    box-shadow: 0 5px 15px rgba(0,0,0,0.1);
}

.изображение-блок img {
```

```
width: 100%;  
display: block;  
border-radius: 5px;  
margin-bottom: 15px;  
}  
  
/* Демонстрационная рамка */  
.демо-рамка {  
background: #f8f9fa;  
padding: 40px;  
border-radius: 10px;  
margin: 30px 0;  
border: 2px dashed #3498db;  
position: relative;  
}  
  
.демо-рамка::before {  
content: "Родительский контейнер";  
position: absolute;  
top: -12px;  
left: 20px;  
background: #3498db;  
color: white;  
padding: 5px 15px;  
border-radius: 20px;  
font-size: 0.9rem;  
}  
  
/* Элемент внутри рамки */  
.демо-элемент {  
background: #2ecc71;  
color: white;  
padding: 20px;  
border-radius: 5px;  
text-align: center;  
font-weight: bold;  
transition: all 0.3s;  
}  
  
/* Интерактивное управление */  
.лаборатория {  
background: #2c3e50;  
color: white;  
padding: 30px;
```

```
border-radius: 15px;
margin: 40px 0;
}

.управление-группа {
    margin-bottom: 20px;
}

label {
    display: block;
    margin-bottom: 8px;
    color: #ecf0f1;
}

select, input {
    width: 100%;
    padding: 10px;
    border-radius: 5px;
    border: none;
    margin-bottom: 15px;
}

.ползунок {
    width: 100%;
}

.кнопка {
    background: #3498db;
    color: white;
    border: none;
    padding: 12px 25px;
    border-radius: 8px;
    cursor: pointer;
    font-size: 16px;
    transition: all 0.3s;
}

.кнопка:hover {
    background: #2980b9;
    transform: translateY(-2px);
}

.результат {
    background: white;
```

```
        color: #333;
        padding: 20px;
        border-radius: 10px;
        margin-top: 20px;
        font-family: 'Courier New', monospace;
    }

/* Адаптивные примеры */
.adaptivnye-primerы {
    display: grid;
    grid-template-columns: repeat(auto-fit, minmax(300px, 1fr));
    gap: 20px;
    margin: 30px 0;
}

.primer-kartochka {
    background: white;
    padding: 25px;
    border-radius: 15px;
    box-shadow: 0 5px 20px rgba(0,0,0,0.1);
}

.primer-kartochka h3 {
    color: #2c3e50;
    margin-bottom: 15px;
    padding-bottom: 10px;
    border-bottom: 2px solid;
}

/* Для демонстрации границ */
.vidimaya-graniца {
    border: 2px solid red !important;
}

</style>
</head>
<body>
    <div class="контейнер" style="max-width: 1000px;">
        <h1>🔥 max-width: «Умная мебель» для вашего сайта</h1>
        <p style="text-align: center; color: #7f8c8d; margin-bottom: 40px;">
            Учимся создавать элементы, которые помещаются в любой «дверной проём»
        </p>
    </div>
    <!-- Раздел 1: Проблема -->
```

```
<div class="раздел">
  <h2>✖ Проблема: «Жёсткая мебель» (только width)</h2>

  <div class="метафора плохая-метафора">
    <h3>📺 Представьте себе:</h3>
    <p>Вы купили огромный телевизор (80 дюймов). Он прекрасно смотрится в
вашей гостиной. Но когда вы пытаетесь:</p>
    <ul style="margin-left: 20px; margin-top: 10px;">
      <li>Перенести его в спальню — он не проходит в дверь</li>
      <li>Показать другу на телефоне — видна только маленькая часть</li>
      <li>Поставить на узкую тумбу — он свисает по краям</li>
    </ul>
    <p style="margin-top: 15px; font-weight: bold; color: #c0392b;">
      Так и с <code>width: 1200px</code> — на телефоне появится
горизонтальная прокрутка!
    </p>
  </div>

  <div class="проблема">
    <div class="контейнер">
      <h3>Контейнер с width: 1200px</h3>
      <p>Попробуйте уменьшить окно браузера. Этот блок останется шириной
1200px и вызовет горизонтальную прокрутку.</p>
      <p>На мобильном телефоне вы увидите только левую часть этого текста и
будете вынуждены скроллить вправо.</p>
    </div>
  </div>
<!-- Раздел 2: Решение -->
<div class="раздел">
  <h2>✓ Решение: «Умная мебель» (max-width)</h2>

  <div class="метафора хорошая-метафора">
    <h3>🚪 Волшебная дверь max-width:</h3>
    <p>Представьте дверь с регулируемой шириной. Она говорит:</p>
    <ul style="margin-left: 20px; margin-top: 10px;">
      <li>«В просторном зале я могу раскрыться на 2 метра»</li>
      <li>«В узком коридоре я сужусь до 80 см»</li>
      <li>«Но никогда не буду шире 2 метров!»</li>
    </ul>
    <p style="margin-top: 15px; font-weight: bold; color: #27ae60;">
      Так работает <code>max-width: 1200px; width: 90%;</code> —
растягивается, но с ограничением!
  </div>
</div>
```

```
</p>
</div>

<div class="решение">
  <div class="контейнер">
    <h3>Контейнер с max-width: 1200px</h3>
    <p>Попробуйте уменьшить окно браузера. Этот блок будет уменьшаться вместе с окном, но никогда не станет шире 1200px.</p>
    <p>На мобильном телефоне весь текст будет удобно читаться без горизонтальной прокрутки!</p>
  </div>
</div>

<!-- Раздел 3: Лаборатория -->
<div class="лаборатория">
  <h2 style="color: white; margin-top: 0;">🔬 Лаборатория max-width</h2>
  <p style="color: #bdc3c7;">Поэкспериментируйте с настройками и посмотрите, как меняется поведение элемента</p>

  <div class="демо-рамка" id="родительский-контейнер">
    <div class="демо-элемент" id="демо-элемент">
      Я — тестируемый элемент. Изменяйте мои настройки справа!
    </div>
  </div>

  <div style="display: grid; grid-template-columns: repeat(auto-fit, minmax(250px, 1fr)); gap: 20px;">
    <div class="управление-группа">
      <label for="тип-ширины">Тип ширины:</label>
      <select id="тип-ширины">
        <option value="width">Только width (проблема)</option>
        <option value="max-width">max-width (решение)</option>
        <option value="оба">width + max-width (лучшее)</option>
      </select>

      <label for="значение-ширины">Значение: <span id="текущее-значение">800</span>px</label>
      <input type="range" class="ползунок" id="значение-ширины" min="200" max="1200" value="800">
    </div>

    <div class="управление-группа">
```

```
<label for="родительская-ширина">Ширина родителя: <span id="родитель-  
значение">100</span>%</label>  
    <input type="range" class="ползунок" id="родительская-ширина" min="30"  
max="100" value="100">  
  
    <label for="показать-границы">  
        <input type="checkbox" id="показать-границы"> Показать границы  
    </label>  
    </div>  
</div>  
  
<div style="text-align: center; margin-top: 20px;">  
    <button class="кнопка" onclick="применитьНастройки()">Применить</button>  
    <button class="кнопка" onclick="сброситьНастройки()" style="background:  
#e74c3c;">Сбросить</button>  
    <button class="кнопка" onclick="показатьКод()" style="background:  
#9b59b6;">Показать CSS-код</button>  
</div>  
  
<div class="результат" id="css-результат">  
    /* Здесь появится CSS-код */  
</div>  
</div>  
  
<!-- Раздел 4: Практические примеры -->  
<div class="раздел">  
    <h2>📱 Практические примеры использования</h2>  
  
    <div class="адаптивные-примеры">  
        <div class="пример-карточка">  
            <h3 style="border-bottom-color: #3498db;">1. Изображения</h3>  
            <p>Самое важное правило для адаптивных изображений:</p>  
            <div class="результат" style="margin-top: 15px; font-size: 0.9rem;">  
                img {<br>  
                    &nbsp;&nbsp;max-width: 100%;<br>  
                    &nbsp;&nbsp;height: auto;<br>  
                }  
            </div>  
            <p style="margin-top: 15px;">Теперь любое изображение никогда не  
выйдет за границы своего контейнера!</p>  
        </div>  
        <div class="пример-карточка">  
            <h3 style="border-bottom-color: #2ecc71;">2. Текстовый контент</h3>
```

```

<p>Для удобства чтения ограничиваем ширину текста:</p>
<div class="результат" style="margin-top: 15px; font-size: 0.9rem;">
    .статья {<br>
        &nbsp;&nbsp;max-width: 800px;<br>
        &nbsp;&nbsp;margin: 0 auto;<br>
        &nbsp;&nbsp;line-height: 1.6;<br>
    }
</div>
<p style="margin-top: 15px;">Строки не становятся слишком длинными на
широких экранах.</p>
</div>

<div class="пример-карточка">
    <h3 style="border-bottom-color: #9b59b6;">3. Основной контейнер</h3>
    <p>Золотой стандарт для контейнеров:</p>
    <div class="результат" style="margin-top: 15px; font-size: 0.9rem;">
        .container {<br>
            &nbsp;&nbsp;max-width: 1200px;<br>
            &nbsp;&nbsp;width: 90%;<br>
            &nbsp;&nbsp;margin: 0 auto;<br>
            &nbsp;&nbsp;padding: 0 15px;<br>
        }
    </div>
    <p style="margin-top: 15px;">На телефоне — 90% ширины, на компьютере
— максимум 1200px.</p>
    </div>
    </div>
</div>

<!-- Раздел 5: Пример с изображениями -->
<div class="раздел">
    <h2> Демонстрация с изображениями</h2>

    <div class="пример-изображений">
        <div class="изображение-блок">
            <h4>✗ Без max-width: 100%</h4>
            <div style="width: 100%; overflow: hidden; border: 2px solid #e74c3c; border-
radius: 5px;">
                
            </div>
        </div>
    </div>
</div>

```

```
<p>На узком экране изображение выходит за границы и вызывает прокрутку.</p>
</div>

<div class="изображение-блок">
  <h4>✓ С max-width: 100%</h4>
  <div style="width: 100%; overflow: hidden; border: 2px solid #2ecc71; border-radius: 5px;">
    
  </div>
  <p>Изображение автоматически уменьшается, чтобы поместиться в контейнер.</p>
</div>
</div>
</div>


<div class="раздел" style="background: #fff8e1; padding: 30px; border-radius: 15px; border-left: 8px solid #f39c12;">
  <h2 style="color: #d35400;">📚 Правила использования max-width</h2>

  <div style="display: grid; grid-template-columns: repeat(auto-fit, minmax(300px, 1fr)); gap: 20px; margin-top: 20px;">
    <div>
      <h4 style="color: #27ae60;">✓ Делайте так:</h4>
      <ul style="margin-left: 20px;">
        <li>Всегда для изображений: <code>max-width: 100%</code></li>
        <li>Для текстовых блоков: <code>max-width: 600-800px</code></li>
        <li>Для основного контейнера: <code>max-width: 1200px; width: 90%</code></li>
        <li>Комбинируйте с <code>margin: 0 auto</code> для центрирования</li>
      </ul>
    </div>
    <div>
      <h4 style="color: #e74c3c;">✗ Не делайте так:</h4>
      <ul style="margin-left: 20px;">
        <li>Не используйте только <code>width</code> без <code>max-width</code></li>
        <li>Не ставьте слишком маленькие значения</li>
      </ul>
    </div>
  </div>
</div>
```

<li>Не забывайте про `height: auto` для изображений</li>

<li>Не используйте для элементов, которые должны быть фиксированными</li>

```
</ul>
</div>
</div>

<div style="margin-top: 30px; padding: 20px; background: white; border-radius: 10px;">
    <h4>💡 Золотая формула для контейнеров:</h4>
    <div class="результат" style="margin-top: 15px;">
        .контейнер {<br>
            &nbsp;&nbsp;<span style="color: #27ae60;">max-width</span>: <span
style="color: #3498db;">1200px</span>;<br>
            &nbsp;&nbsp;<span style="color: #27ae60;">width</span>: <span style="color:
#3498db;">90%</span>;<br>
            &nbsp;&nbsp;<span style="color: #9b59b6;">margin</span>: <span
style="color: #3498db;">0 auto</span>;<br>
            &nbsp;&nbsp;<span style="color: #9b59b6;">padding</span>: <span
style="color: #3498db;">0 15px</span>;<br>
        }
    </div>
    <p style="margin-top: 15px;">Эта комбинация работает идеально на любых
устройствах!</p>
    </div>
</div>

<!-- Раздел 7: Домашнее задание -->
<div class="раздел" style="background: linear-gradient(135deg, #667eea 0%, #764ba2 100%); color: white; padding: 30px; border-radius: 15px;">
    <h2 style="color: white;">📝 Домашнее задание</h2>

    <div style="display: grid; grid-template-columns: repeat(auto-fit, minmax(250px, 1fr)); gap: 20px; margin-top: 20px;">
        <div style="background: rgba(255,255,255,0.1); padding: 20px; border-radius: 10px;">
            <h4>Задание 1: Адаптивная статья</h4>
            <p>Создайте страницу статьи с:</p>
            <ul>
                <li>Контейнером <code>max-width: 800px</code></li>
                <li>Изображениями с <code>max-width: 100%</code></li>
                <li>Центрированием через <code>margin: 0 auto</code></li>
            </ul>
        </div>
    </div>
```

```

<div style="background: rgba(255,255,255,0.1); padding: 20px; border-radius: 10px;">
    <h4>Задание 2: Карточки товаров</h4>
    <p>Создайте сетку карточек:</p>
    <ul>
        <li>Карточка: <code>max-width: 300px</code></li>
        <li>Контейнер: <code>max-width: 1200px</code></li>
        <li>Адаптивность: 4/3/2/1 карточки в ряд</li>
    </ul>
</div>

<div style="background: rgba(255,255,255,0.1); padding: 20px; border-radius: 10px;">
    <h4>Задание 3: Сравнение</h4>
    <p>Создайте две одинаковые страницы:</p>
    <ul>
        <li>С <code>width: 1200px</code></li>
        <li>С <code>max-width: 1200px</code></li>
        <li>Сравните на телефоне и компьютере</li>
    </ul>
</div>
</div>

<div style="text-align: center; margin-top: 25px;">
    <button class="кнопка" onclick="alert('Удачи в выполнении заданий! Помните: max-width — ваш лучший друг в адаптивном дизайне!')">
        style="background: white; color: #764ba2; font-weight: bold;">
            Принять задания!
        </button>
    </div>
</div>
</div>

<script>
    // Получаем элементы
    const демоЭлемент = document.getElementById('демо-элемент');
    const родительскийКонтейнер = document.getElementById('родительский-контейнер');
    const типШирины = document.getElementById('тип-ширины');
    const значениеШирины = document.getElementById('значение-ширины');
    const текущееЗначение = document.getElementById('текущее-значение');
    const родительскаяШирина = document.getElementById('родительская-ширина');
    const родительЗначение = document.getElementById('родитель-значение');
    const показатьГраницы = document.getElementById('показать-границы');

```

```
const cssРезультат = document.getElementById('css-результат');

// Функция применения настроек
function применитьНастройки() {
    const тип = типШирины.value;
    const значение = значениеШирины.value + 'px';
    const родительШирина = родительскаяШирина.value + '%';

    // Обновляем отображение значений
    текущееЗначение.textContent = значениеШирины.value;
    родительЗначение.textContent = родительскаяШирина.value;

    // Применяем ширину родителя
    родительскийКонтейнер.style.width = родительШирина;

    // Применяем настройки к демо-элементу
    демоЭлемент.style.width = '';
    демоЭлемент.style.maxWidth = '';

    switch(тип) {
        case 'width':
            демоЭлемент.style.width = значение;
            демоЭлемент.textContent = `width: ${значение} (проблема на узких
экранах)`;

            break;

        case 'max-width':
            демоЭлемент.style.maxWidth = значение;
            демоЭлемент.textContent = `max-width: ${значение} (умное решение)`;
            break;

        case 'оба':
            демоЭлемент.style.width = '100%';
            демоЭлемент.style.maxWidth = значение;
            демоЭлемент.textContent = `width: 100%; max-width: ${значение}
(идеально!)`;

            break;
    }

    // Показываем/скрываем границы
    if (показатьГраницы.checked) {
        демоЭлемент.classList.add('видимая-граница');
        родительскийКонтейнер.classList.add('видимая-граница');
    } else {

```

```

демоЭлемент.classList.remove('видимая-граница');
родительскийКонтейнер.classList.remove('видимая-граница');
}

// Обновляем CSS-код
обновитьCSSКод(тип, значение, родительШирина);

// Показываем подсказку
if (тип === 'width' && parseInt(родительскаяШирина.value) < 50) {
    setTimeout(() => {
        alert('⚠️ Обратите внимание: при узком родителе и фиксированном width
элемент выходит за границы!');
    }, 300);
}

// Функция обновления CSS-кода
function обновитьCSSКод(тип, значение, родительШирина) {
    let код = "";

    switch(тип) {
        case 'width':
            код = `.демо-элемент {\n    width: ${значение};\n    /* Проблема: на узком
экране выйдет за границы */\n}`;
            break;

        case 'max-width':
            код = `.демо-элемент {\n    max-width: ${значение};\n    /* Решение: никогда
не будет шире ${значение} */\n}`;
            break;

        case 'оба':
            код = `.демо-элемент {\n    width: 100%;\n    max-width: ${значение};\n    /* Идеально: растягивается, но с ограничением */\n}`;
            break;
    }

    код += `\n\n.родительский-контейнер {\n    width: ${родительШирина};\n}`;

    cssРезультат.textContent = код;
}

// Функция сброса настроек
function сброситьНастройки() {

```

```
типыШирины.value = 'max-width';
значениеШирины.value = 800;
родительскаяШирина.value = 100;
показатьГраницы.checked = false;
применитьНастройки();

демоЭлемент.textContent = 'Я — тестируемый элемент. Изменяйте мои настройки
справа!';
}

// Функция показа кода
function показатьКод() {
    const код = cssРезультат.textContent;
    alert('Скопируйте этот CSS-код:\n\n' + код);
}

// Назначаем обработчики
значениеШирины.addEventListener('input', function() {
    текущееЗначение.textContent = this.value;
});

родительскаяШирина.addEventListener('input', function() {
    родительЗначение.textContent = this.value;
});

// Инициализация
сброситьНастройки();

// Демонстрация изменения размера окна
let размерОкна = 100;
let направление = -1;

setInterval(() => {
    if (демоЭлемент.textContent.includes('умное решение') ||
        демоЭлемент.textContent.includes('идеально')) {
        размерОкна += направление * 5;
        if (размерОкна <= 40 || размерОкна >= 100) направление *= -1;
        родительскийКонтейнер.style.width = размерОкна + '%';
        родительскаяШирина.value = размерОкна;
        родительЗначение.textContent = размерОкна;
    }
}, 500);
</script>
</body>
```

</html>

---



## Сравнительная таблица: width vs max-width

Ситуация	Только width	max-width	width + max-width
Широкий экран	Фиксированная ширина	Растёт до предела	Растёт до предела
Узкий экран	Выходит за границы, прокрутка	Уменьшается, помещается	Уменьшается, помещается
Контроль	Полный, но жёсткий	Гибкий, с ограничением сверху	Идеальный баланс
Адаптивность	Низкая	Высокая	Максимальная
Пример	width: 1200px	max-width: 1200px	width: 100%; max-width: 1200px



## Ключевые выводы:

### 1. Всегда используйте для изображений:

css

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

### 2. Для контейнеров — золотой стандарт:

css

```
.container {  
    max-width: 1200px; /* Максимум */  
    width: 90%;      /* На узких экранах */  
    margin: 0 auto;   /* Центрирование */  
}
```

### 3. Для текста — оптимальная читаемость:

css

```
.article {  
    max-width: 800px; /* 50-75 символов в строке */  
    margin: 0 auto;  
    line-height: 1.6;  
}
```

### 4. Комбинируйте с min-width для полного контроля:

css

```
.element {  
    min-width: 300px; /* Не узче 300px */  
    max-width: 800px; /* Не шире 800px */  
    width: 100%;      /* Всё доступное пространство между */  
}
```

---



#### Простые правила на каждый день:

1. **Думайте: «Максимум, но не больше»** — это философия max-width
  2. **Начинайте с max-width, а не с width**
  3. **Тестируйте на телефоне** — уменьшайте окно браузера во время разработки
  4. **Изображения = max-width: 100%** — запомните это как мантру
  5. **Центрируйте через margin: 0 auto** — чтобы ограниченный контент был по центру
- 



#### Главный секрет адаптивного дизайна:

max-width — это не ограничение, а свобода.

Свобода для элемента расти на больших экранах и сжиматься на маленьких, всегда оставаясь удобным и красивым.

Теперь ваша «мебель» (элементы сайта) будет помещаться в любой «дверной проём» (размер экрана)! От огромного телевизора до маленького смартфона — ваш сайт будет выглядеть идеально.

## Часть 3: Магия превращений. Делаем страницу живой и удобной

- Глава 7: «Умный» способ применять стили – КЛАССЫ.
  - Зачем плодить одинаковые инструкции? Даем элементам «имена» (классы).

Дорогой друг! Представьте, что вы ведёте кулинарную книгу. У вас есть рецепт «Борщ». Вы записываете его один раз, а потом просто говорите: «Приготовьте борщ!» — и все знают, что нужно делать. В CSS классы работают точно так же! Это «имена» для элементов, которые позволяют применять одни и те же стили ко множеству элементов без повторения кода.

---

### Проблема: «Копипастный ад»

#### Ситуация без классов:

Вы создаёте сайт садовода с тремя блоками:

```
css

/* Блок про розы */
#розы {
    background: #ffebee;
    padding: 20px;
    border-radius: 10px;
    border-left: 5px solid #e74c3c;
    margin-bottom: 20px;
}

/* Блок про тюльпаны */
#тюльпаны {
    background: #ffebee;      /* То же самое! */
    padding: 20px;            /* То же самое! */
    border-radius: 10px;       /* То же самое! */
```

```
border-left: 5px solid #e74c3c; /* То же самое! */
margin-bottom: 20px;      /* То же самое! */
}

/* Блок про лилии */
#лилии {
    background: #ffebee;      /* И снова то же самое! */
    padding: 20px;
    border-radius: 10px;
    border-left: 5px solid #e74c3c;
    margin-bottom: 20px;
}
```

## Проблемы:

- Много повторяющегося кода** — 15 строк вместо 5
  - Сложно изменять** — чтобы изменить отступы, нужно править 3 места
  - Легко ошибиться** — можно забыть изменить один из блоков
  - Файл растёт** — больше кода = медленнее загрузка
- 



## Решение: Классы – «умные имена»

### С классами мы делаем так:

css

```
/* ОПИСЫВАЕМ ОДИН РАЗ */
.цветочный-блок {
    background: #ffebee;
    padding: 20px;
    border-radius: 10px;
    border-left: 5px solid #e74c3c;
    margin-bottom: 20px;
}
```

/\* И ПРИМЕНЯЕМ МНОГО РАЗ \*/

html

```
<div class="цветочный-блок">
    <h3>Розы</h3>
    <p>Красные, розовые, белые...</p>
</div>
```

```
<div class="цветочный-блок">
    <h3>Тюльпаны</h3>
    <p>Весенние цветы...</p>
</div>
```

```
<div class="цветочный-блок">
    <h3>Лилии</h3>
    <p>Элегантные и ароматные...</p>
</div>
```

### Преимущества:

1. **Один раз описали** — много раз используем
  2. **Легко изменять** — поменяли класс, изменились все элементы
  3. **Чистый код** — нет повторений
  4. **Логичная структура** — сразу видно, какие элементы одинаковые
- 

## 🎯 Что такое классы в CSS?

**Класс** — это «имя», которое мы даём элементу, чтобы применять к нему определённые стили.

### Правила создания классов:

1. **Начинается с точки** в CSS: .название-класса
2. **Может содержать:** буквы, цифры, дефисы, подчёркивания
3. **Не может начинаться с цифры**
4. **Чувствителен к регистру** (но лучше использовать один стиль)

css

```
/* Правильно */
.красивый-блок {
    color: red;
}

.красивыйБлок {
    color: blue;
}
```

```
.блок_с_подчёркиванием {
    color: green;
}

/* Неправильно */
.123блок {      /* Нельзя начинать с цифры */
    color: red;
}

.блок с пробелом { /* Нельзя использовать пробелы */
    color: blue;
}
```

---

## Как создать и использовать классы?

### Шаг 1: Создаём класс в CSS

css

```
/* Это наш "рецепт" для важных объявлений */
.важное-объявление {
    background: #fff3cd; /* Жёлтый фон */
    border: 2px solid #ffc107; /* Оранжевая рамка */
    border-radius: 8px;
    padding: 15px;
    margin: 20px 0;
    font-weight: bold;
    color: #856404;
}
```

### Шаг 2: Применяем класс в HTML

html

```
<!-- Применяем наш класс к div -->
<div class="важное-объявление">
    ! Внимание! Завтра полив цветов в 8 утра.
</div>
```

```
<!-- Применяем тот же класс к другому div -->
<div class="важное-объявление">
```

 Специальное предложение: скидка 20% на семена роз!  
</div>

<!-- Даже к параграфу можно применить -->  
<p class="важное-объявление">  
  Не забудьте: собрание садоводов в субботу.  
</p>

**Результат:** Все три элемента получат одинаковое оформление!

---

## Практический пример: Создаём систему стилей

Создайте файл `классы-пример.html`:

html

```
<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Волшебство классов: один стиль — много элементов</title>
    <style>
        /* ===== БАЗОВЫЕ СТИЛИ ===== */
        * {
            box-sizing: border-box;
        }

        body {
            font-family: 'Arial', sans-serif;
            line-height: 1.6;
            color: #333;
            background: #f9f9f9;
            padding: 20px;
            max-width: 1000px;
            margin: 0 auto;
        }

        h1, h2 {
            color: #2c3e50;
            text-align: center;
        }
    
```

```
}

/* ===== НАШИ КЛАССЫ ===== */

/* 1. Класс для уведомлений */
.уведомление {
    padding: 15px 20px;
    border-radius: 8px;
    margin: 15px 0;
    border-left: 5px solid;
}

/* 2. Разные типы уведомлений (на основе общего класса) */
.уведомление.информация {
    background: #d1ecf1;
    border-left-color: #17a2b8;
    color: #0c5460;
}

.уведомление.успех {
    background: #d4edda;
    border-left-color: #28a745;
    color: #155724;
}

.уведомление.предупреждение {
    background: #fff3cd;
    border-left-color: #ffc107;
    color: #856404;
}

.уведомление.ошибка {
    background: #f8d7da;
    border-left-color: #dc3545;
    color: #721c24;
}

/* 3. Класс для карточек */
.карточка {
    background: white;
    border-radius: 12px;
    padding: 25px;
    margin: 20px 0;
    box-shadow: 0 5px 15px rgba(0,0,0,0.08);
}
```

```
        transition: transform 0.3s;
    }

    .карточка:hover {
        transform: translateY(-5px);
        box-shadow: 0 10px 25px rgba(0,0,0,0.12);
    }

/* 4. Класс для кнопок */
.knopka {
    display: inline-block;
    padding: 12px 25px;
    border-radius: 6px;
    text-decoration: none;
    font-weight: bold;
    cursor: pointer;
    border: none;
    transition: all 0.3s;
    margin: 5px;
}

.knopka.основная {
    background: #3498db;
    color: white;
}

.knopka.основная:hover {
    background: #2980b9;
    transform: translateY(-2px);
    box-shadow: 0 5px 15px rgba(52, 152, 219, 0.3);
}

.knopka.вторичная {
    background: #95a5a6;
    color: white;
}

.knopka.успех {
    background: #2ecc71;
    color: white;
}

.knopka.опасность {
    background: #e74c3c;
```

```
        color: white;
    }

/* 5. Класс для цитат */
.цитата {
    font-style: italic;
    padding: 20px;
    background: #f8f9fa;
    border-radius: 8px;
    border-left: 5px solid #3498db;
    margin: 20px 0;
}

.цитата .автор {
    text-align: right;
    font-weight: bold;
    color: #7f8c8d;
    margin-top: 10px;
}

/* 6. Класс для подсказок */
.подсказка {
    background: #e8f4fc;
    padding: 15px;
    border-radius: 8px;
    border: 2px dashed #3498db;
    margin: 20px 0;
    font-size: 0.9rem;
}

/* 7. Класс для галереи фотографий */
.фото-галерея {
    display: flex;
    flex-wrap: wrap;
    gap: 15px;
    margin: 20px 0;
}

.фото {
    flex: 1;
    min-width: 200px;
    height: 150px;
    border-radius: 8px;
    overflow: hidden;
}
```

```
background: linear-gradient(45deg, #667eea, #764ba2);
display: flex;
align-items: center;
justify-content: center;
color: white;
font-weight: bold;
}

/* ===== ДЕМОНСТРАЦИОННЫЕ СТИЛИ ===== */
.демо-секция {
    background: white;
    padding: 30px;
    border-radius: 15px;
    margin: 30px 0;
    box-shadow: 0 5px 20px rgba(0,0,0,0.1);
}

.код-пример {
    background: #2c3e50;
    color: #ecf0f1;
    padding: 15px;
    border-radius: 8px;
    font-family: 'Courier New', monospace;
    margin: 15px 0;
    overflow-x: auto;
}

.лаборатория {
    background: #f8f9fa;
    padding: 25px;
    border-radius: 10px;
    margin: 25px 0;
    border: 2px solid #e9ecef;
}

</style>
</head>
<body>
    <h1>🧙‍♂️ Магия классов в CSS</h1>
    <p style="text-align: center; color: #7f8c8d;">Один раз опиши стиль — используй много раз!</p>

    <!-- Демонстрационная секция 1: Уведомления -->
    <div class="демо-секция">
        <h2>1. Уведомления (один класс — четыре стиля)</h2>
```

```
<div class="уведомление информация">  
     Информация: Сегодня солнечно, идеальный день для полива цветов.  
</div>
```

```
<div class="уведомление успех">  
     Успех: Ваши розы успешно пересажены! Полив завтра утром.  
</div>
```

```
<div class="уведомление предупреждение">  
     Предупреждение: Завтра возможен дождь, не забудьте укрыть рассаду.  
</div>
```

```
<div class="уведомление ошибка">  
     Ошибка: Не удалось добавить удобрение. Проверьте срок годности.  
</div>
```

```
<div class="подсказка">  
     Все эти уведомления используют один базовый класс  
<code>.уведомление</code>  
    плюс дополнительный класс (<code>.информация</code>, <code>.успех</code> и  
т.д.)  
</div>
```

```
<div class="код-пример">  
    &lt;!-- HTML --><br>  
    &lt;div class="уведомление информация"&gt;...&lt;/div&gt;<br>  
    &lt;div class="уведомление успех"&gt;...&lt;/div&gt;<br><br>  
  
    /* CSS */<br>  
    .уведомление {<br>  
        &nbsp;&nbsp; padding: 15px 20px;<br>  
        &nbsp;&nbsp; border-radius: 8px;<br>  
    }<br><br>  
  
    .уведомление.информация {<br>  
        &nbsp;&nbsp; background: #d1ecf1;<br>  
        &nbsp;&nbsp; border-left-color: #17a2b8;<br>  
    }  
</div>  
</div>
```

```
<!-- Демонстрационная секция 2: Карточки -->  
<div class="демо-секция">  
    <h2>2. Карточки (одинаковый стиль для разного контента)</h2>
```

```
<div style="display: grid; grid-template-columns: repeat(auto-fit, minmax(300px, 1fr)); gap: 20px;">

    <div class="карточка">
        <h3>🌹 Уход за розами</h3>
        <p>Розы требуют регулярного полива, подкормки и обрезки. Лучшее время для полива — утро.</p>
        <button class="кнопка основная">Подробнее</button>
    </div>

    <div class="карточка">
        <h3>☀️ Выращивание подсолнухов</h3>
        <p>Подсолнухи любят солнце и хорошо дренированную почву. Высаживайте их на солнечной стороне.</p>
        <button class="кнопка вторичная">Читать далее</button>
    </div>

    <div class="карточка">
        <h3>🌸 Весенние луковичные</h3>
        <p>Тюльпаны, нарциссы и гиацинты высаживают осенью для цветения весной.</p>
        <button class="кнопка успех">Купить луковицы</button>
    </div>

</div>

<div class="подсказка">
     Все карточки используют класс .карточка. Если мы захотим изменить тень или отступы, нам нужно изменить код только в одном месте!
</div>
</div>

<!-- Демонстрационная секция 3: Кнопки --&gt;
&lt;div class="демо-секция"&gt;
    &lt;h2&gt;3. Кнопки (система стилей)&lt;/h2&gt;

    &lt;p&gt;Разные кнопки, но общие правила:&lt;/p&gt;
    &lt;div style="margin: 20px 0; padding: 20px; background: #f8f9fa; border-radius: 8px;"&gt;
        &lt;button class="кнопка основная"&gt;Основная кнопка&lt;/button&gt;
        &lt;button class="кнопка вторичная"&gt;Вторичная кнопка&lt;/button&gt;
        &lt;button class="кнопка успех"&gt;Кнопка успеха&lt;/button&gt;
        &lt;button class="кнопка опасность"&gt;Опасная кнопка&lt;/button&gt;
        &lt;a href="#" class="кнопка основная"&gt;Ссылка-кнопка&lt;/a&gt;
    &lt;/div&gt;</pre>
```

```
<div class="подсказка">
     Кнопки используют общий класс <code>.кнопка</code> (отступы, скругления, переходы)
    и дополнительные классы для цвета (<code>.основная</code>, <code>.успех</code>
    и т.д.)
</div>
</div>
```

*<!-- Демонстрационная секция 4: Цитаты -->*

```
<div class="демо-секция">
```

```
    <h2>4. Цитаты (единий стиль оформления)</h2>
```

```
    <div class="цитата">
```

"Сад — это великий учитель. Он учит терпению и бережному наблюдению. Он учит трудолюбию и экономии. Больше всего он учит полному доверию."

```
        <div class="автор">— Гертруда Джекилл</div>
```

```
    </div>
```

```
    <div class="цитата">
```

"Цветы всегда делают людей лучше, счастливее и полезнее; они — солнечный свет, пища и медицина для души."

```
        <div class="автор">— Лютер Бербанк</div>
```

```
    </div>
```

```
    <div class="код-пример">
```

```
        &lt;!-- Все цитаты используют один класс --&gt;<br>
```

```
        &lt;div class="цитата"&gt;<br>
```

```
            &nbsp;&nbsp;"Текст цитаты..."<br>
```

```
            &nbsp;&nbsp;&lt;div class="автор"&gt;— Автор&lt;/div&gt;<br>
```

```
        &lt;/div&gt;
```

```
    </div>
```

```
</div>
```

*<!-- Лаборатория: Создаём свой класс -->*

```
<div class="лаборатория">
```

```
    <h2> Лаборатория: Создайте свой класс!</h2>
```

```
    <p>Давайте создадим класс для специальных предложений:</p>
```

```
<div style="display: grid; grid-template-columns: 1fr 1fr; gap: 20px;">
```

```
    <div>
```

```
        <h3>Настройки класса:</h3>
```

```
<label>Фон:
    <select id="фон">
        <option value="#fff3cd">Жёлтый</option>
        <option value="#d4edda">Зелёный</option>
        <option value="#d1ecf1">Голубой</option>
        <option value="#f8d7da">Розовый</option>
    </select>
</label>

<label>Цвет рамки:
    <input type="color" id="рамка-цвет" value="#ffc107">
</label>

<label>Толщина рамки:
    <input type="range" id="рамка-толщина" min="1" max="10" value="3">
    <span id="толщина-значение">3</span>px
</label>

<label>Скругление углов:
    <input type="range" id="скругление" min="0" max="20" value="10">
    <span id="скругление-значение">10</span>px
</label>

<button onclick="создатьКласс()" style="background: #3498db; color: white; border: none; padding: 10px 20px; border-radius: 5px; cursor: pointer; margin-top: 10px;">
    Создать класс
</button>
</div>

<div>
    <h3>Результат:</h3>
    <div id="результат-блок" style="padding: 20px; margin: 10px 0; border: 3px solid #ffc107; border-radius: 10px; background: #fff3cd;">
        Ваше специальное предложение появится здесь!
    </div>
</div>

<div id="css-код" class="код-пример" style="margin-top: 20px;">
    /* CSS-код появится здесь */
</div>
</div>
</div>

<div style="margin-top: 20px;">
    <h4>Примеры использования вашего класса:</h4>
```

```
<div id="примеры-использования" style="display: flex; gap: 10px; flex-wrap: wrap; margin-top: 10px;">
    <!-- Здесь появятся примеры -->
</div>
</div>
</div>

<!-- Секция сравнения: с классами и без -->
<div class="демо-секция" style="background: #2c3e50; color: white;">
    <h2 style="color: white;">⚖ Сравнение: с классами и без</h2>

    <div style="display: grid; grid-template-columns: 1fr 1fr; gap: 20px; margin-top: 20px;">
        <div>
            <h3 style="color: #e74c3c;">✗ Без классов</h3>
            <div class="код-пример">
                #блок1 {<br>
                    &nbsp;&nbsp;background: #fff3cd;<br>
                    &nbsp;&nbsp;padding: 20px;<br>
                    &nbsp;&nbsp;border-radius: 10px;<br>
                }<br><br>

                #блок2 {<br>
                    &nbsp;&nbsp;background: #fff3cd; /* Повтор! */<br>
                    &nbsp;&nbsp;padding: 20px; /* Повтор! */<br>
                    &nbsp;&nbsp;border-radius: 10px; /* Повтор! */<br>
                }<br><br>

                #блок3 {<br>
                    &nbsp;&nbsp;background: #fff3cd; /* И снова повтор! */<br>
                    &nbsp;&nbsp;padding: 20px;<br>
                    &nbsp;&nbsp;border-radius: 10px;<br>
                }
            </div>
            <p><strong>Проблемы:</strong> 9 строк кода, много повторений, сложно
изменять.</p>
        </div>
    </div>
    <div>
        <h3 style="color: #2ecc71;">✓ С классами</h3>
        <div class="код-пример">
            .специальный-блок {<br>
                &nbsp;&nbsp;background: #fff3cd;<br>
                &nbsp;&nbsp;padding: 20px;<br>
                &nbsp;&nbsp;border-radius: 10px;<br>
            }
        </div>
    </div>

```

```
}<br><br>

<!-- HTML --><br>
<div id="блок1" class="специальный-блок">...</div><br>
<div id="блок2" class="специальный-блок">...</div><br>
<div id="блок3" class="специальный-блок">...</div>
</div>
<p><strong>Преимущества:</strong> 3 строки кода, нет повторений, легко
изменять.</p>
</div>
</div>
</div>

<!-- Итоги и домашнее задание -->
<div class="карточка" style="background: linear-gradient(135deg, #667eea 0%, #764ba2 100%); color: white;">
    <h2 style="color: white;">🎓 Итоги и домашнее задание</h2>

    <div style="display: grid; grid-template-columns: repeat(auto-fit, minmax(250px, 1fr)); gap: 20px; margin-top: 20px;">
        <div style="background: rgba(255,255,255,0.1); padding: 20px; border-radius: 10px;">
            <h4>Что мы узнали:</h4>
            <ul>
                <li>Классы начинаются с точки: <code>.имя-класса</code></li>
                <li>Один класс можно применять ко многим элементам</li>
                <li>Элемент может иметь несколько классов</li>
                <li>Классы экономят код и упрощают поддержку</li>
            </ul>
        </div>
    </div>

    <div style="background: rgba(255,255,255,0.1); padding: 20px; border-radius: 10px;">
        <h4>Домашнее задание:</h4>
        <p><strong>Создайте систему классов для сайта садовода:</strong></p>
        <ol>
            <li>Класс <code>.растение-карточка</code></li>
            <li>Класс <code>.совет-садовода</code></li>
            <li>Класс <code>.сезонная-акция</code></li>
            <li>Используйте каждый класс минимум 3 раза</li>
        </ol>
    </div>
</div>

<div style="text-align: center; margin-top: 25px;">
```

```
<button onclick="alert('Удачи в создании классов! Помните: хороший класс — как  
хороший рецепт: один раз создал, много раз использовал.')" style="background: white; color: #764ba2; border: none; padding: 12px 25px;  
border-radius: 8px; cursor: pointer; font-weight: bold;">  
    Начать создавать классы!  
</button>  
</div>  
</div>  
  
<script>  
    // Лаборатория: Создание класса  
    function создатьКласс() {  
        const фон = document.getElementById('фон').value;  
        const рамкаЦвет = document.getElementById('рамка-цвет').value;  
        const рамкаТолщина = document.getElementById('рамка-толщина').value;  
        const скругление = document.getElementById('скругление').value;  
  
        // Обновляем значения  
        document.getElementById('толщина-значение').textContent = рамкаТолщина;  
        document.getElementById('скругление-значение').textContent = скругление;  
  
        // Применяем стили к демо-блоку  
        const результатБлок = document.getElementById('результат-блок');  
        результатБлок.style.background = фон;  
        результатБлок.style.border = `${рамкаТолщина}px solid ${рамкаЦвет}`;  
        результатБлок.style.borderRadius = `${скругление}px`;  
        результатБлок.innerHTML = `  
            <h3 style="margin-top: 0;">☀ Специальное предложение!</h3>  
            <p>Только сегодня скидка 30% на все семена цветов. Успейте купить!</p>  
            <button class="кнопка успех" style="margin-top: 10px;">Купить со  
            скидкой</button>  
        `;  
  
        // Генерируем CSS-код  
        const cssКод = document.getElementById('css-код');  
        cssКод.innerHTML =  
            `.специальное-предложение {  
                background: ${фон};  
                border: ${рамкаТолщина}px solid ${рамкаЦвет};  
                border-radius: ${скругление}px;  
                padding: 20px;  
                margin: 15px 0;  
            }`;  
    };
```

```

// Создаём примеры использования
const примерыИспользования = document.getElementById('примеры-
использования');
примерыИспользования.innerHTML =
    <div class="карточка" style="border: ${рамкаТолщина}px solid ${рамкаЦвет}; border-radius: ${скругление}px; background: ${фон};">
        <small>Карточка товара</small>
        <h4 style="margin: 10px 0;">Семена роз</h4>
        <p>299 ₽ <s>399 ₽</s></p>
    </div>

    <div class="карточка" style="border: ${рамкаТолщина}px solid ${рамкаЦвет}; border-radius: ${скругление}px; background: ${фон};">
        <small>Новость</small>
        <h4 style="margin: 10px 0;">Новый сорт тюльпанов</h4>
        <p>Поступление на следующей неделе</p>
    </div>

    <div class="карточка" style="border: ${рамкаТолщина}px solid ${рамкаЦвет}; border-radius: ${скругление}px; background: ${фон};">
        <small>Напоминание</small>
        <h4 style="margin: 10px 0;">Полив завтра</h4>
        <p>Не забудьте полить розы в 8 утра</p>
    </div>
    ;
}

// Инициализация
document.getElementById('рамка-толщина').addEventListener('input', function() {
    document.getElementById('толщина-значение').textContent = this.value;
});

document.getElementById('скругление').addEventListener('input', function() {
    document.getElementById('скругление-значение').textContent = this.value;
});

создатьКласс();

// Демонстрация добавления класса
setTimeout(() => {
    const кнопки = document.querySelectorAll('.кнопка');
    кнопки.forEach(кнопка => {
        кнопка.classList.add('с-анимацией');
    });
}

```

```

// Добавляем стиль для анимации
const стиль = document.createElement('style');
стиль.textContent = `
    .с-анимацией {
        animation: pulse 2s infinite;
    }

    @keyframes pulse {
        0% { transform: scale(1); }
        50% { transform: scale(1.05); }
        100% { transform: scale(1); }
    }
};

document.head.appendChild(стиль);
}, 2000);
</script>
</body>
</html>

```

---



## Сравнение подходов: ID vs Классы

Критерий	ID (#)	Классы (.)
<b>Синтаксис</b>	#уникальное-имя	.имя-класса
<b>Уникальность</b>	Один на странице	Много на странице
<b>Использование</b>	Уникальные элементы (шапка, подвал)	Повторяющиеся элементы (карточки, кнопки)
<b>Пример</b>	#основной-заголовок	.красивая-кнопка
<b>В HTML</b>	id="имя"	class="имя"
<b>Приоритет</b>	Высокий	Средний



## Ключевые преимущества классов:

### 1. Повторное использование

css

```
/* Один раз создали */
.красивый-блок {
    /* стили */
}

/* Много раз используем */
<div class="красивый-блок">...</div>
<div class="красивый-блок">...</div>
<div class="красивый-блок">...</div>
```

### 2. Комбинация классов

html

```
<!-- Элемент может иметь несколько классов -->
<div class="карточка специальное-предложение новинка">
    ...
</div>
```

css

```
/* Каждый класс добавляет свои стили */
.карточка { /* общие стили карточки */ }
.специальное-предложение { /* жёлтый фон */ }
.новинка { /* красная метка "NEW" */ }
```

### 3. Легкость изменений

Хотите изменить все карточки на сайте? Просто отредактируйте класс `.карточка`!

### 4. Читаемость кода

html

```
<!-- По классам сразу понятно назначение -->
<div class="статья главная-новость">...</div>
<button class="кнопка купить большая">Купить</button>
```

```
<div class="уведомление предупреждение">...</div>
```

---



## Практические советы по именованию классов:

### Хорошие имена:

css

```
/* Описывают назначение */
```

```
.меню-навигации  
.кнопка-отправки  
.карточка-товара  
.заголовок-раздела
```

```
/* Описывают состояние */
```

```
.активная-вкладка  
.скрытый-элемент  
.заблокированная-кнопка
```

```
/* Используют БЭМ (популярная методология) */
```

```
.блок__элемент  
.блок--модификатор
```

### Плохие имена:

css

```
/* Слишком общие */
```

```
.красный  
.большой  
.правый
```

```
/* Описывают внешний вид (а что если дизайн изменится?) */
```

```
.синяя-рамка /* Завтра рамка станет зелёной */  
.крупный-текст /* На телефоне текст будет слишком крупным */
```

```
/* Непонятные сокращения */
```

```
.mnu /* menu? */  
.btn-prm /* primary button? */
```

---



## Золотые правила работы с классами:

1. «**Один раз опиши — много раз используй**» — основная философия
  2. **Используйте осмысленные имена** — по названию сразу должно быть понятно назначение
  3. **Комбинируйте классы** — базовый класс + модификаторы
  4. **Избегайте !important** — если нужно переопределить класс, создайте более специфичный
  5. **Документируйте сложные классы** — комментарии в CSS помогут вам и другим
  6. **Создавайте систему** — последовательные имена для всей системы
- 



## Главный секрет профессионального CSS:

**Классы — это «словарь» вашего сайта.**

Хорошо подобранные имена классов делают код понятным, поддерживаемым и масштабируемым.

Инвестируйте время в создание хорошей системы классов — и ваш код будет благодарить вас годами!

Теперь вы умеете создавать «умные» стили с помощью классов! В следующей главе мы научимся создавать простое меню навигации — гордость любого сайта, используя всё, что мы уже изучили.

## ■ Как создать класс в CSS (точка перед названием) и «присвоить» его в HTML.

Дорогой друг! Сейчас мы разберём самый важный практический навык в CSS — создание и использование классов. Это как научиться готовить по рецепту: один раз записали рецепт (создали класс), а потом много раз готовите по нему (применяете к элементам). Давайте сделаем это шаг за шагом!

---

### 🎯 Шаг 1: Понимаем метафору «Кулинарная книга»

Представьте, что **CSS-файл** — это ваша кулинарная книга с рецептами.

Каждый **класс** — это один рецепт.

А **HTML-элементы** — это блюда, которые вы готовите по этим рецептам.

text

КУЛИНАРНАЯ КНИГА (CSS-файл)

- └── Рецепт "Борщ" (класс .борщ)
- └── Рецепт "Пельмени" (класс .пельмени)
- └── Рецепт "Компот" (класс .компот)

НА КУХНЕ (HTML-страница)

- └── Блюдо 1: Приготовлено по рецепту "Борщ" (<div class="борщ">)
  - └── Блюдо 2: Приготовлено по рецепту "Пельмени" (<div class="пельмени">)
  - └── Блюдо 3: Приготовлено по рецепту "Компот" (<div class="компот">)
- 

### 🔧 Шаг 2: Создаём класс в CSS (пишем рецепт)

**Правило №1: Класс всегда начинается с точки .**

В CSS мы объявляем класс так:

css

```
.имя-класса {  
    свойство: значение;
```

```
        другое-свойство: значение;  
    }
```

## Пример 1: Простой класс для кнопки

css

```
/* В CSS файле (style.css) */  
.моя-кнопка {  
    background-color: #3498db; /* Синий фон */  
    color: white;           /* Белый текст */  
    padding: 12px 24px;     /* Отступы внутри */  
    border: none;           /* Без рамки */  
    border-radius: 6px;      /* Закруглённые углы */  
    font-size: 16px;         /* Размер текста */  
    cursor: pointer;        /* Курсор-рука при наведении */  
}
```

## Пример 2: Класс для карточки с товаром

css

```
/* В CSS файле (style.css) */  
.товарная-карточка {  
    width: 300px;           /* Ширина карточки */  
    background: white;       /* Белый фон */  
    border-radius: 12px;      /* Закруглённые углы */  
    padding: 20px;           /* Внутренние отступы */  
    box-shadow: 0 5px 15px rgba(0,0,0,0.1); /* Лёгкая тень */  
    margin: 15px;            /* Внешние отступы */  
    transition: transform 0.3s; /* Плавная анимация */  
}  
  
/* Эффект при наведении на карточку */  
.товарная-карточка:hover {  
    transform: translateY(-5px); /* Поднимаем карточку */  
    box-shadow: 0 10px 25px rgba(0,0,0,0.15); /* Усиливаем тень */  
}
```

## Пример 3: Класс для уведомлений

css

```
/* В CSS файле (style.css) */  
.уведомление {  
    padding: 15px 20px;  
    border-radius: 8px;
```

```
margin: 15px 0;  
border-left: 5px solid /* Цветная полоса слева */;  
font-weight: bold;  
}  
  
/* Разные типы уведомлений */  
.уведомление.успех {  
background: #d4edda;  
border-left-color: #28a745;  
color: #155724;  
}  
  
.уведомление.предупреждение {  
background: #fff3cd;  
border-left-color: #ffc107;  
color: #856404;  
}  
  
.уведомление.ошибка {  
background: #f8d7da;  
border-left-color: #dc3545;  
color: #721c24;  
}
```

---



## Шаг 3: Присваиваем класс в HTML (готовим по рецепту)

### Правило №2: В HTML класс указывается без точки!

В HTML мы используем атрибут `class`:

```
html  
<элемент class="имя-класса">...</элемент>
```

#### Пример 1: Применяем класс кнопки

```
html  
<!-- В HTML файле (index.html) -->  
<button class="моя-кнопка">Нажми меня!</button>
```

```
<button class="моя-кнопка">Купить сейчас</button>
<button class="моя-кнопка">Добавить в корзину</button>
```

**Результат:** Все три кнопки получат одинаковые стили из класса .моя-кнопка

## Пример 2: Применяем класс карточки

html

```
<!-- В HTML файле (index.html) -->
<div class="товарная-карточка">
    <h3>Чайник электрический</h3>
    <p>Мощность: 2200 Вт</p>
    <p>Цена: 2 499 ₽</p>
    <button class="моя-кнопка">Купить</button>
</div>

<div class="товарная-карточка">
    <h3>Блендер погружной</h3>
    <p>Мощность: 800 Вт</p>
    <p>Цена: 3 199 ₽</p>
    <button class="моя-кнопка">Купить</button>
</div>
```

## Пример 3: Применяем классы уведомлений

html

```
<!-- В HTML файле (index.html) -->
<div class="уведомление успех">
     Заказ успешно оформлен! Доставка завтра.
</div>
```

```
<div class="уведомление предупреждение">
     Товар заканчивается на складе.
</div>
```

```
<div class="уведомление ошибка">
     Не удалось добавить товар в корзину.
</div>
```

---

## Шаг 4: Практикум «Создаём систему классов для сайта садовода»

Давайте создадим полноценный пример с нуля! Создайте три файла:

### Файл 1: index.html (структура)

```
html

<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Сад моей мечты - Классы в действии</title>
    <!-- Подключаем наш CSS файл -->
    <link rel="stylesheet" href="style.css">
    <!-- Подключаем шрифты -->
    <link rel="preconnect" href="https://fonts.googleapis.com">
    <link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
    <link href="https://fonts.googleapis.com/css2?
family=Playfair+Display:wght@400;700&family=Raleway:wght@300;400;600&display=swap"
rel="stylesheet">
</head>
<body>
    <!-- Контейнер сайта -->
    <div class="контейнер">

        <!-- Шапка сайта -->
        <header class="шапка-сайта">
            <h1 class="главный-заголовок">☀ Сад моей мечты</h1>
            <p class="подзаголовок">Практический пример создания и использования CSS-
классов</p>
        </header>

        <!-- Навигационное меню -->
        <nav class="меню-навигации">
            <a href="#" class="пункт-меню активный">Главная</a>
            <a href="#" class="пункт-меню">Каталог растений</a>
            <a href="#" class="пункт-меню">Советы садовода</a>
            <a href="#" class="пункт-меню">О нас</a>
            <a href="#" class="пункт-меню">Контакты</a>
        </nav>
    </div>
</body>
```

```
</nav>

<!-- Основное содержимое --&gt;
&lt;main class="основное-содержимое"&gt;

    <!-- Секция с уведомлениями --&gt;
    &lt;section class="секция"&gt;
        &lt;h2 class="заголовок-секции"&gt;📢 Уведомления (пример классов)&lt;/h2&gt;

        &lt;div class="уведомление информация"&gt;
            <img alt="Lightbulb icon" data-bbox="298 268 318 283"/> <strong>Информация:</strong> Сегодня идеальный день для посадки роз.
        </div>

        <div class="уведомление успех">
             <strong>Успех:</strong> Ваши тюльпаны успешно взошли!
        </div>

        <div class="уведомление предупреждение">
             <strong>Предупреждение:</strong> Завтра возможны заморозки, укройте растения.
        </div>
    </section>

    <!-- Секция с карточками растений --&gt;
    &lt;section class="секция"&gt;
        &lt;h2 class="заголовок-секции"&gt;🌿 Наши растения (один класс - много карточек)&lt;/h2&gt;

        &lt;div class="сетка-карточек"&gt;
            &lt;!-- Карточка 1 --&gt;
            &lt;div class="карточка-растения"&gt;
                &lt;div class="изображение-растения" style="background: linear-gradient(45deg, #667eea, #764ba2);"&gt;
                    <img alt="Rose flower icon" data-bbox="382 738 398 753"/>
                </div>
                <div class="содержимое-карточки">
                    <h3 class="название-растения">Роза "Красная мечта"</h3>
                    <p class="описание-растения">
                        Ароматная красная роза, цветёт с июня по сентябрь.  

                        Любит солнечные места и регулярный полив.
                    </p>
                    <div class="цена-блок">
                        <span class="цена">450 ₽</span>
                    </div>
                </div>
            </div>
        </div>
    </section>

```

```
        <button class="кнопка кнопка-купить">В корзину</button>
    </div>
</div>

<!-- Карточка 2 -->
<div class="карточка-растения">
    <div class="изображение-растения" style="background: linear-gradient(45deg, #43e97b, #38f9d7);">
        
    </div>
    <div class="содержимое-карточки">
        <h3 class="название-растения">Тюльпан "Весенний  
рассвет"</h3>
        <p class="описание-растения">
            Нежно-розовые тюльпаны, цветут в апреле-мае.  

            Идеальны для весенних клумб и букетов.
        </p>
        <div class="цена-блок">
            <span class="цена">280 ₽</span>
            <button class="кнопка кнопка-купить">В корзину</button>
        </div>
    </div>
</div>

<!-- Карточка 3 -->
<div class="карточка-растения">
    <div class="изображение-растения" style="background: linear-gradient(45deg, #fa709a, #fee140);">
        
    </div>
    <div class="содержимое-карточки">
        <h3 class="название-растения">Сакура "Японская вишня"</h3>
        <p class="описание-растения">
            Декоративное дерево с нежными розовыми цветами.  

            Цветёт в мае, создавая волшебную атмосферу.
        </p>
        <div class="цена-блок">
            <span class="цена">1 890 ₽</span>
            <button class="кнопка кнопка-купить">В корзину</button>
        </div>
    </div>
</div>
</div>
```

```
</section>

<!-- Секция с советами -->
<section class="секция">
    <h2 class="заголовок-секции"> Советы садовода (разные классы)</h2>

    <div class="совет-блок важно">
        <h4 class="заголовок-совета"> Правильный полив</h4>
        <p>Поливайте растения утром или вечером, когда солнце не активно.  
Это предотвратит ожоги листьев и испарение воды.</p>
    </div>

    <div class="совет-блок полезно">
        <h4 class="заголовок-совета"> Подготовка почвы</h4>
        <p>Перед посадкой добавьте в почву компост или перегной.  
Это обеспечит растения питательными веществами на весь сезон.</p>
    </div>

    <div class="совет-блок интересно">
        <h4 class="заголовок-совета"> Привлечение пчёл</h4>
        <p>Посадите лаванду, шалфей и мяту рядом с овощными грядками.  
Эти растения привлекают пчёл для опыления.</p>
    </div>
</section>

<!-- Лаборатория классов -->
<section class="секция лаборатория-классов">
    <h2 class="заголовок-секции"> Лаборатория: Создайте свой класс!</h2>

    <div class="инструкция">
        <p>Измените параметры ниже и создайте свой уникальный класс для  
карточек:</p>
    </div>

    <div class="настройки-класса">
        <div class="настройка">
            <label>Фон карточки:</label>
            <input type="color" id="цвет-фона" value="#ffffff">
        </div>

        <div class="настройка">
            <label>Цвет рамки:</label>
            <input type="color" id="цвет-рамки" value="#3498db">
        </div>
    </div>

```

```
<div class="настройка">
    <label>Толщина рамки: <span id="толщина-
значение">3</span>px</label>
        <input type="range" id="толщина-рамки" min="1" max="10" value="3">
    </div>

    <div class="настройка">
        <label>Скругление углов: <span id="скругление-
значение">12</span>px</label>
            <input type="range" id="скругление-углов" min="0" max="30"
value="12">
        </div>

        <button class="кнопка кнопка-создать" onclick="создатьКласс()">
            Создать класс
        </button>
    </div>

    <div class="результат-лаборатории">
        <h4>Ваш класс в действии:</h4>
        <div class="демо-карточка" id="демо-карточка">
            <h3>Пример карточки</h3>
            <p>Это демонстрация вашего класса. Все карточки с этим классом
будут выглядеть так!</p>
        </div>
    </div>

    <div class="код-блок">
        <h4>CSS-код вашего класса:</h4>
        <pre id="css-код"><code>.мой-класс-карточка {
/* Ваш код появится здесь */
}</code></pre>
    </div>
</div>
</section>
</main>


<footer class="подвал-сайта">
    <p class="текст-подвала">© 2023 Сад моей мечты. Все классы защищены 😊</p>
    <p class="текст-подвала маленький">Изучайте CSS с удовольствием! Один раз
создал класс — много раз использовал.</p>
</footer>
</div>
```

```
<!-- Подключаем JavaScript -->
<script src="скрипт.js"></script>
</body>
</html>
```

## Файл 2: style.css (все наши классы)

css

```
/* ===== */
/* ОСНОВНЫЕ НАСТРОЙКИ И БАЗОВЫЕ КЛАССЫ */
/* ===== */

/* Сброс отступов по умолчанию */
* {
    box-sizing: border-box;
    margin: 0;
    padding: 0;
}

/* Класс для основного контейнера */
.контейнер {
    max-width: 1200px;      /* Максимальная ширина */
    width: 90%;            /* На узких экранах 90% */
    margin: 0 auto;         /* Центрирование */
    padding: 20px;
    font-family: 'Raleway', 'Arial', sans-serif;
    line-height: 1.6;
    color: #333;
    background-color: #f9f9f9;
    min-height: 100vh;
}

/* ===== */
/* КЛАССЫ ДЛЯ ТИПОГРАФИИ */
/* ===== */

/* Класс для главного заголовка */
.главный-заголовок {
    font-family: 'Playfair Display', 'Georgia', serif;
    font-size: 2.5rem;      /* Крупный размер */
    color: #2c3e50;         /* Тёмно-синий цвет */
    text-align: center;     /* Выравнивание по центру */
```

```
margin-bottom: 10px;      /* Отступ снизу */
text-shadow: 1px 1px 3px rgba(0,0,0,0.1); /* Лёгкая тень */
}

/* Класс для подзаголовка */
.подзаголовок {
    text-align: center;      /* По центру */
    color: #7f8c8d;         /* Серо-голубой цвет */
    font-size: 1.2rem;       /* Чуть крупнее обычного текста */
    margin-bottom: 30px;     /* Отступ снизу */
    font-weight: 300;        /* Лёгкое начертание */
}

/* Класс для заголовков секций */
.заголовок-секции {
    color: #2c3e50;          /* Тёмно-синий */
    font-size: 1.8rem;        /* Размер */
    margin-bottom: 25px;      /* Отступ снизу */
    padding-bottom: 10px;      /* Отступ для линии */
    border-bottom: 3px solid #3498db; /* Синяя линия снизу */
    font-family: 'Playfair Display', serif;
}

/* ===== */
/* КЛАССЫ ДЛЯ СТРУКТУРЫ САЙТА */
/* ===== */

/* Класс для шапки сайта */
.шапка-сайта {
    background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);
    padding: 40px;
    border-radius: 15px;
    margin-bottom: 30px;
    color: white;
    text-align: center;
    box-shadow: 0 10px 30px rgba(0,0,0,0.15);
}

/* Класс для навигационного меню */
.меню-навигации {
    display: flex;           /* Элементы в строку */
    flex-wrap: wrap;         /* Перенос на новую строку */
    gap: 10px;               /* Промежутки между элементами */
    background: white;       /* Белый фон */
}
```

```
padding: 15px;           /* Внутренние отступы */
border-radius: 10px;     /* Закруглённые углы */
margin-bottom: 30px;     /* Отступ снизу */
box-shadow: 0 5px 15px rgba(0,0,0,0.08); /* Тень */
justify-content: center; /* Центрирование элементов */

}

/* Класс для пунктов меню */
.пункт-меню {
    padding: 10px 20px;   /* Отступы внутри */
    text-decoration: none; /* Убираем подчёркивание */
    color: #2c3e50;       /* Цвет текста */
    border-radius: 5px;    /* Закруглённые углы */
    transition: all 0.3s; /* Плавные переходы */
    font-weight: 600;      /* Полужирный текст */
}

/* Класс для активного пункта меню */
.пункт-меню.активный {
    background: #3498db; /* Синий фон */
    color: white;        /* Белый текст */
}

/* Эффект при наведении на пункт меню */
.пункт-меню:hover {
    background: #f8f9fa; /* Светлый фон */
    transform: translateY(-2px); /* Лёгкий подъём */
}

/* Класс для основного содержимого */
.основное-содержимое {
    background: white;    /* Белый фон */
    padding: 30px;         /* Внутренние отступы */
    border-radius: 15px;   /* Закруглённые углы */
    box-shadow: 0 5px 20px rgba(0,0,0,0.1); /* Тень */
    margin-bottom: 30px;   /* Отступ снизу */
}

/* Класс для секций */
.секция {
    margin-bottom: 40px;   /* Отступ между секциями */
    padding-bottom: 30px;   /* Отступ снизу */
    border-bottom: 1px solid #eee; /* Серая разделительная линия */
}
```

```
/* Убираем границу у последней секции */
.секция:last-child {
    border-bottom: none;
    margin-bottom: 0;
    padding-bottom: 0;
}

/* Класс для подвала сайта */
.подвал-сайта {
    text-align: center; /* Выравнивание по центру */
    padding: 20px; /* Внутренние отступы */
    color: #7f8c8d; /* Серо-голубой цвет */
    font-size: 0.9rem; /* Немного мельче основного текста */
    background: white; /* Белый фон */
    border-radius: 10px; /* Закруглённые углы */
    box-shadow: 0 5px 15px rgba(0,0,0,0.08); /* Лёгкая тень */
}

/* Класс для текста в подвале */
.текст-подвала {
    margin-bottom: 10px; /* Отступ снизу */
}

/* Класс для мелкого текста */
.текст-подвала.маленький {
    font-size: 0.8rem; /* Ещё мельче */
    color: #95a5a6; /* Светло-серый */
}

/* ===== */
/* КЛАССЫ ДЛЯ УВЕДОМЛЕНИЙ */
/* ===== */

/* БАЗОВЫЙ КЛАСС для всех уведомлений */
.уведомление {
    padding: 15px 20px; /* Внутренние отступы */
    border-radius: 8px; /* Закруглённые углы */
    margin: 15px 0; /* Внешние отступы */
    border-left: 5px solid; /* Цветная полоса слева */
    font-weight: bold; /* Жирный текст */
    display: flex; /* Для иконки и текста */
    align-items: center; /* Вертикальное выравнивание */
    gap: 10px; /* Промежуток между иконкой и текстом */
}
```

```
}

/* Класс для информационных уведомлений */
.уведомление.информация {
    background: #d1ecf1;      /* Светло-голубой фон */
    border-left-color: #17a2b8; /* Бирюзовая полоса */
    color: #0c5460;          /* Тёмно-синий текст */
}
```

```
/* Класс для уведомлений об успехе */
.уведомление.успех {
    background: #d4edda;      /* Светло-зелёный фон */
    border-left-color: #28a745; /* Зелёная полоса */
    color: #155724;          /* Тёмно-зелёный текст */
}
```

```
/* Класс для предупреждений */
.уведомление.предупреждение {
    background: #fff3cd;      /* Светло-жёлтый фон */
    border-left-color: #ffc107; /* Жёлтая полоса */
    color: #856404;          /* Тёмно-жёлтый текст */
}
```

```
=====
/* КЛАССЫ ДЛЯ КАРТОЧЕК РАСТЕНИЙ */
=====
```

```
/* Класс-контейнер для сетки карточек */
.сетка-карточек {
    display: grid;           /* Используем CSS Grid */
    grid-template-columns: repeat(auto-fit, minmax(300px, 1fr)); /* Адаптивные колонки */
    gap: 25px;               /* Промежутки между карточками */
    margin: 20px 0;           /* Отступы сверху и снизу */
}
```

```
/* ОСНОВНОЙ КЛАСС для карточек растений */
.карточка-растения {
    background: white;        /* Белый фон */
    border-radius: 12px;       /* Закруглённые углы */
    overflow: hidden;         /* Обрезаем выходящее за границы */
    box-shadow: 0 5px 15px rgba(0,0,0,0.1); /* Лёгкая тень */
    transition: all 0.3s ease; /* Плавные переходы */
    display: flex;            /* Flexbox для макета */
    flex-direction: column;    /* Вертикальное расположение */
}
```

```
}

/* Эффект при наведении на карточку */
.карточка-растения:hover {
    transform: translateY(-10px);      /* Поднимаем карточку */
    box-shadow: 0 15px 35px rgba(0,0,0,0.15); /* Усиливаем тень */
}

/* Класс для изображения растения */
.изображение-растения {
    height: 200px;                  /* Фиксированная высота */
    display: flex;                 /* Для центрирования иконки */
    align-items: center;           /* Вертикальное центрирование */
    justify-content: center;       /* Горизонтальное центрирование */
    font-size: 4rem;               /* Крупная иконка */
    color: white;                 /* Белый цвет иконки */
}

/* Класс для содержимого карточки */
.содержимое-карточки {
    padding: 25px;                /* Внутренние отступы */
    flex-grow: 1;                  /* Занимает всё доступное пространство */
    display: flex;                /* Flexbox */
    flex-direction: column;         /* Вертикальное расположение */
}

/* Класс для названия растения */
.название-растения {
    color: #2c3e50;               /* Тёмно-синий цвет */
    margin-bottom: 10px;            /* Отступ снизу */
    font-size: 1.3rem;              /* Размер текста */
    font-family: 'Playfair Display', serif;
}

/* Класс для описания растения */
.описание-растения {
    color: #7f8c8d;               /* Серо-голубой цвет */
    line-height: 1.5;                /* Межстрочный интервал */
    margin-bottom: 20px;              /* Отступ снизу */
    flex-grow: 1;                  /* Занимает всё доступное пространство */
}

/* Класс для блока с ценой */
.цена-блок {
```

```
display: flex;           /* Flexbox для цены и кнопки */
justify-content: space-between; /* Цена слева, кнопка справа */
align-items: center;      /* Вертикальное выравнивание */
margin-top: 20px;         /* Отступ сверху */
padding-top: 20px;        /* Отступ сверху внутри блока */
border-top: 1px solid #eee; /* Разделительная линия */
}

/* Класс для цены */
.цена {
    font-size: 1.5rem;      /* Крупный размер */
    font-weight: bold;       /* Жирное начертание */
    color: #27ae60;         /* Зелёный цвет */
}

/* ===== */
/* КЛАССЫ ДЛЯ КНОПОК */
/* ===== */

/* БАЗОВЫЙ КЛАСС для всех кнопок */
.кнопка {
    padding: 10px 20px;     /* Внутренние отступы */
    border-radius: 6px;      /* Закруглённые углы */
    border: none;           /* Без рамки */
    font-size: 1rem;         /* Размер текста */
    font-weight: 600;        /* Полужирное начертание */
    cursor: pointer;        /* Курсор-рука при наведении */
    transition: all 0.3s;    /* Плавные переходы */
    display: inline-block;   /* Чтобы работали отступы */
    text-align: center;      /* Выравнивание текста */
}

/* Класс для кнопки "Купить" */
.кнопка.кнопка-купить {
    background: #2ecc71;    /* Зелёный фон */
    color: white;           /* Белый текст */
}

/* Эффект при наведении на кнопку "Купить" */
.кнопка.кнопка-купить:hover {
    background: #27ae60;    /* Тёмно-зелёный фон */
    transform: translateY(-2px); /* Лёгкий подъём */
    box-shadow: 0 5px 15px rgba(46, 204, 113, 0.3); /* Зелёная тень */
}
```

```
/* Класс для кнопки "Создать" (в лаборатории) */
.кнопка.кнопка-создать {
    background: #3498db;          /* Синий фон */
    color: white;                /* Белый текст */
    padding: 12px 30px;           /* Более крупные отступы */
    font-size: 1.1rem;            /* Чуть крупнее текст */
    margin-top: 20px;             /* Отступ сверху */
}

/* Эффект при наведении на кнопку "Создать" */
.кнопка.кнопка-создать:hover {
    background: #2980b9;          /* Тёмно-синий фон */
    transform: translateY(-2px);   /* Лёгкий подъём */
    box-shadow: 0 5px 15px rgba(52, 152, 219, 0.3); /* Синяя тень */
}

/* ===== */
/* КЛАССЫ ДЛЯ СОВЕТОВ САДОВОДА */
/* ===== */

/* БАЗОВЫЙ КЛАСС для всех советов */
.совет-блок {
    padding: 20px;                /* Внутренние отступы */
    border-radius: 10px;            /* Закруглённые углы */
    margin-bottom: 20px;           /* Отступ снизу */
    border-left: 5px solid;         /* Цветная полоса слева */
}

/* Класс для важных советов */
.совет-блок.важно {
    background: #e8f4fc;           /* Светло-голубой фон */
    border-left-color: #3498db;      /* Синяя полоса */
}

/* Класс для полезных советов */
.совет-блок.полезно {
    background: #e8f5e9;           /* Светло-зелёный фон */
    border-left-color: #2ecc71;      /* Зелёная полоса */
}

/* Класс для интересных советов */
.совет-блок.интересно {
    background: #f9f0ff;            /* Светло-фиолетовый фон */
```

```
border-left-color: #9b59b6;      /* Фиолетовая полоса */
}

/* Класс для заголовка совета */
.заголовок-совета {
    color: #2c3e50;            /* Тёмно-синий цвет */
    margin-bottom: 10px;        /* Отступ снизу */
    font-size: 1.2rem;          /* Размер текста */
    display: flex;              /* Для иконки и текста */
    align-items: center;        /* Вертикальное выравнивание */
    gap: 10px;                 /* Промежуток между иконкой и текстом */
}

/*
===== */

/* КЛАССЫ ДЛЯ ЛАБОРАТОРИИ */
/* ===== */

/* Класс для лаборатории */
.лаборатория-классов {
    background: #f8f9fa;        /* Светло-серый фон */
    padding: 30px;              /* Внутренние отступы */
    border-radius: 15px;         /* Закруглённые углы */
    border: 2px dashed #3498db;  /* Синяя пунктирная рамка */
}

/* Класс для инструкции в лаборатории */
.инструкция {
    margin-bottom: 20px;        /* Отступ снизу */
    color: #7f8c8d;             /* Серо-голубой цвет */
    font-size: 1.1rem;           /* Чуть крупнее обычного */
}

/* Класс для настроек класса */
.настройки-класса {
    display: grid;              /* Используем CSS Grid */
    grid-template-columns: repeat(auto-fit, minmax(200px, 1fr)); /* Адаптивные колонки */
    gap: 15px;                  /* Промежутки между элементами */
    margin-bottom: 30px;         /* Отступ снизу */
    background: white;           /* Белый фон */
    padding: 25px;               /* Внутренние отступы */
    border-radius: 10px;          /* Закруглённые углы */
    box-shadow: 0 5px 15px rgba(0,0,0,0.08); /* Лёгкая тень */
}
```

```
/* Класс для отдельной настройки */
.настройка {
    display: flex;          /* Flexbox для label и input */
    flex-direction: column;   /* Вертикальное расположение */
    gap: 8px;                /* Промежуток между элементами */
}

/* Стили для label в настройках */
.настройка label {
    font-weight: 600;        /* Полужирное начертание */
    color: #2c3e50;         /* Тёмно-синий цвет */
}

/* Стили для input в настройках */
.настройка input {
    padding: 10px;           /* Внутренние отступы */
    border: 2px solid #ddd;   /* Серая рамка */
    border-radius: 5px;       /* Закруглённые углы */
    font-size: 1rem;          /* Размер текста */
    transition: border-color 0.3s; /* Плавное изменение рамки */
}

/* Эффект при фокусе на input */
.настройка input:focus {
    border-color: #3498db;    /* Синяя рамка при фокусе */
    outline: none;            /* Убираем стандартный outline */
}

/* Класс для результата лаборатории */
.результат-лаборатории {
    background: white;        /* Белый фон */
    padding: 25px;             /* Внутренние отступы */
    border-radius: 10px;        /* Закруглённые углы */
    box-shadow: 0 5px 15px rgba(0,0,0,0.08); /* Лёгкая тень */
}

/* Класс для демонстрационной карточки */
.демо-карточка {
    padding: 30px;             /* Внутренние отступы */
    border-radius: 12px;        /* Закруглённые углы */
    margin: 20px 0;             /* Отступы сверху и снизу */
    background: white;          /* Белый фон по умолчанию */
    border: 3px solid #3498db;  /* Синяя рамка по умолчанию */
}
```

```
/* Класс для блока с кодом */
.код-блок {
    margin-top: 20px;           /* Отступ сверху */
}

/* Стили для pre (блок с кодом) */
.код-блок pre {
    background: #2c3e50;        /* Тёмно-синий фон */
    color: #ecf0f1;            /* Светло-серый текст */
    padding: 20px;              /* Внутренние отступы */
    border-radius: 8px;         /* Закруглённые углы */
    overflow-x: auto;           /* Горизонтальная прокрутка при необходимости */
    font-family: 'Courier New', monospace; /* Монотипичный шрифт */
    font-size: 0.9rem;          /* Чуть мельче обычного текста */
    margin-top: 10px;           /* Отступ сверху */
}

/* Стили для code внутри pre */
.код-блок code {
    font-family: inherit;       /* Наследуем монотипичный шрифт */
}

/* ===== */
/* АДАПТИВНОСТЬ ДЛЯ МОБИЛЬНЫХ УСТРОЙСТВ */
/* ===== */

@media (max-width: 768px) {
    /* На мобильных уменьшаем размер шрифтов */
    .главный-заголовок {
        font-size: 2rem;
    }

    .подзаголовок {
        font-size: 1rem;
    }

    .заголовок-секции {
        font-size: 1.5rem;
    }

    /* Меню навигации становится вертикальным */
    .меню-навигации {
        flex-direction: column;      /* Вертикальное расположение */
    }
}
```

```
    gap: 5px;          /* Уменьшаем промежутки */
}

.пункт-меню {
    text-align: center; /* Выравнивание текста по центру */
}

/* Сетка карточек становится одно- или двухколоночной */
.сетка-карточек {
    grid-template-columns: 1fr; /* Одна колонка */
}

/* Настройки класса становятся вертикальными */
.настройки-класса {
    grid-template-columns: 1fr; /* Одна колонка */
}

/* Уменьшаем отступы */
.основное-содержимое {
    padding: 20px;
}

.секция {
    padding: 20px 0;
}

}

@media (max-width: 480px) {
    /* На очень маленьких экранах ещё уменьшаем отступы */
    .контейнер {
        width: 95%;
        padding: 10px;
    }

    .шапка-сайта {
        padding: 20px;
    }

    .демо-карточка {
        padding: 20px;
    }
}

/* ===== */
```

```

/* ДОПОЛНИТЕЛЬНЫЕ ЭФФЕКТЫ */
/* ===== */

/* Плавное появление элементов */
.секция, .карточка-растения, .совет-блок {
    animation: fadeIn 0.5s ease-out;
}

@keyframes fadeIn {
    from {
        opacity: 0;
        transform: translateY(20px);
    }
    to {
        opacity: 1;
        transform: translateY(0);
    }
}

/* Эффект пульсации для кнопок при наведении */
.кнопка:hover {
    animation: pulse 0.5s;
}

@keyframes pulse {
    0% {
        transform: scale(1);
    }
    50% {
        transform: scale(1.05);
    }
    100% {
        transform: scale(1);
    }
}

```

## Файл 3: скрипт.js (интерактивность для лаборатории)

```

javascript

// Ждём, пока вся страница загрузится
document.addEventListener('DOMContentLoaded', function() {
    // Инициализируем лабораторию
    инициализироватьЛабораторию();
})

```

```
// Добавляем интерактивность кнопкам "В корзину"
добавитьОбработчикиКнопок();

});

// Функция для инициализации лаборатории
function инициализироватьЛабораторию() {
    // Получаем элементы управления
    const цветФона = document.getElementById('цвет-фона');
    const цветРамки = document.getElementById('цвет-рамки');
    const толщинаРамки = document.getElementById('толщина-рамки');
    const скруглениеУглов = document.getElementById('скругление-углов');

    // Получаем элементы для отображения значений
    const толщинаЗначение = document.getElementById('толщина-значение');
    const скруглениеЗначение = document.getElementById('скругление-значение');

    // Обновляем значения при изменении ползунков
    толщинаРамки.addEventListener('input', function() {
        толщинаЗначение.textContent = this.value;
    });

    скруглениеУглов.addEventListener('input', function() {
        скруглениеЗначение.textContent = this.value;
    });

    // Создаём класс при загрузке страницы
    создатьКласс();
}

// Функция для создания класса
function создатьКласс() {
    // Получаем значения из элементов управления
    const цветФона = document.getElementById('цвет-фона').value;
    const цветРамки = document.getElementById('цвет-рамки').value;
    const толщинаРамки = document.getElementById('толщина-рамки').value;
    const скруглениеУглов = document.getElementById('скругление-углов').value;

    // Применяем стили к демонстрационной карточке
    const демоКарточка = document.getElementById('демо-карточка');
    демоКарточка.style.background = цветФона;
    демоКарточка.style.border = `${толщинаРамки}px solid ${цветРамки}`;
    демоКарточка.style.borderRadius = `${скруглениеУглов}px`;
```

```
// Генерируем CSS-код
const cssКод = document.getElementById('css-код');
const код = `.мой-класс-карточка {
background: ${цветФона};
border: ${толщинаРамки}px solid ${цветРамки};
border-radius: ${скруглениеУглов}px;
padding: 30px;
margin: 20px 0;
box-shadow: 0 5px 15px rgba(0,0,0,0.1);
transition: all 0.3s ease;
}

/* Эффект при наведении */
.мой-класс-карточка:hover {
    transform: translateY(-5px);
    box-shadow: 0 10px 25px rgba(0,0,0,0.15);
};

// Вставляем код в блок
cssКод.innerHTML = код;

// Показываем сообщение о успешном создании класса
показатьУведомление(`Класс создан! Используйте <span style="color: #3498db; font-weight: bold;">.мой-класс-карточка</span> в вашем CSS.`);
}

// Функция для показа уведомления
function показатьУведомление(сообщение) {
    // Создаём элемент уведомления
    const уведомление = document.createElement('div');
    уведомление.className = 'уведомление информация';
    уведомление.style.position = 'fixed';
    уведомление.style.top = '20px';
    уведомление.style.right = '20px';
    уведомление.style.zIndex = '1000';
    уведомление.style.maxWidth = '300px';
    уведомление.innerHTML = `🎨 ${сообщение}`;

    // Добавляем уведомление на страницу
    document.body.appendChild(уведомление);

    // Удаляем уведомление через 3 секунды
    setTimeout(() => {
        уведомление.style.opacity = '0';
    
```

```
уведомление.style.transform = 'translateX(100%)';
	setTimeout(() => {
		document.body.removeChild(уведомление);
	}, 300);
}, 3000);

}

// Функция для добавления обработчиков к кнопкам "В корзину"
function добавитьОбработчикиКнопок() {
	const кнопкиКупить = document.querySelectorAll('.кнопка-купить');

	кнопкиКупить.forEach(кнопка => {
		кнопка.addEventListener('click', function(event) {
			// Предотвращаем стандартное поведение
			event.preventDefault();

			// Получаем название растения из карточки
			const карточка = this.closest('.карточка-растения');
			const название = карточка.querySelector('.название-растения').textContent;
			const цена = карточка.querySelector('.цена').textContent;

			// Сохраняем оригинальный текст кнопки
			const оригиналныйТекст = this.textContent;

			// Меняем текст кнопки
			this.textContent = '✓ Добавлено!';
			this.style.background = '#27ae60';

			// Показываем уведомление
			показатьУведомление(`"${название}" добавлен в корзину за ${цена}`);

			// Возвращаем оригинальный текст через 2 секунды
			setTimeout(() => {
				this.textContent = оригиналныйТекст;
				this.style.background = '';
			}, 2000);
		});
	});
}

// Добавляем глобальную функцию для кнопки "Создать класс"
window.создатьКласс = создатьКласс;
```

---



# Пошаговое руководство: От создания до применения класса

## Шаг 1: Придумайте имя класса

- Хорошо: .красивая-кнопка, .карточка-товара, .важное-уведомление
- Плохо: .красная (описывает цвет, а не назначение), .большая (описывает размер)

## Шаг 2: Создайте класс в CSS

css

```
/* В файле style.css */  
.ваш-класс {  
    /* ваши свойства здесь */  
    color: blue;  
    font-size: 18px;  
    margin: 10px;  
}
```

## Шаг 3: Примените класс в HTML

html

```
<!-- В файле index.html -->  
<div class="ваш-класс">Этот элемент получит стили из класса</div>  
<p class="ваш-класс">Этот параграф тоже получит те же стили!</p>  
<button class="ваш-класс">И эта кнопка тоже</button>
```

## Шаг 4: Проверьте результат

Откройте файл index.html в браузере и убедитесь, что все элементы с классом выглядят одинаково.

---



## Важные правила и хитрости:

### 1. Один элемент — много классов

html

```
<!-- Можно применять несколько классов к одному элементу -->
<div class="карточка специальное-предложение новинка">
    Эта карточка получит стили из всех трёх классов!
</div>
```

### 2. Классы можно комбинировать в CSS

css

```
/* Стили применяются только к элементам с ОБОИМИ классами */
.карточка.специальное-предложение {
    border: 3px solid gold;
}
```

### 3. Наследование стилей

css

```
/* Стили применяются ко всем элементам ВНУТРИ класса */
.карточка p {
    color: #666; /* Все параграфы внутри .карточка будут серыми */
}

.карточка .цена {
    font-weight: bold; /* Все элементы с классом .цена внутри .карточка */
}
```

### 4. Приоритеты

Если к элементу применено несколько конфликтующих стилей, браузер использует:

1. !important (избегайте!)
2. Стиль в атрибуте style="..."
3. Класс с большей специфичностью
4. Класс, объявленный позже в CSS

---

## Практическое задание:

### Задание 1: Создайте систему классов для кулинарного сайта

1. Класс .рецепт-карточка для карточек рецептов
2. Класс .ингредиент для списка ингредиентов
3. Класс .шаг-приготовления для шагов приготовления
4. Класс .совет-повара для полезных советов

### Задание 2: Создайте классы для разных типов кнопок

1. .кнопка-основная — для главных действий
2. .кнопка-вторичная — для дополнительных действий
3. .кнопка-успех — для подтверждающих действий
4. .кнопка-опасность — для удаления/отмены

### Задание 3: Эксперимент с наследованием

Создайте класс .красная-рамка и примените его к разным элементам:

html

```
<div class="красная-рамка">Див с красной рамкой</div>
<p class="красная-рамка">Абзац с красной рамкой</p>

```

---

## Ключевые выводы:

1. Классы в CSS начинаются с точки: .имя-класса
2. Классы в HTML указываются без точки: class="имя-класса"
3. Один класс можно применять ко многим элементам
4. Один элемент может иметь несколько классов
5. Имена классов должны описывать назначение, а не внешний вид



## Философия классов:

**Классы — это как слова в языке CSS.**

Хорошо подобранные имена классов делают ваш код понятным, поддерживаемым и масштабируемым.

Чем лучше вы назовёте класс сегодня, тем легче вам будет работать с кодом завтра!

**Поздравляю!** Теперь вы умеете создавать и использовать CSS-классы — один из самых важных навыков в веб-разработке. В следующей главе мы создадим простое меню навигации, используя всё, что мы уже изучили о классах!

## ■ Практикум №5: Создаем стиль для «особо важного объявления» и «изящной подписи».

Дорогой друг! Пришло время для настоящего волшебства. Мы научились раскрашивать текст, выбирать шрифты и выравнивать его. Теперь давайте соберём все эти знания воедино и создадим два особых стиля, которые украсят любую вашу страницу:

1. **«Особо важное объявление»** — чтобы его невозможно было пропустить!
2. **«Изящная подпись»** — чтобы завершить письмо или статью с достоинством и вкусом.

Это как вязать две разные, но одинаково красивые детали одного изделия: одна — яркая и заметная, другая — тонкая и изысканная.

---

## ШАГ 1: Создаём нашу рабочую папку и файлы

Давайте начнём с чистого листа. Создайте на Рабочем столе папку **Практикум5\_СпецСтили**.

Внутри этой папки создайте два файла:

1. **index.html** — наша веб-страница с примером
  2. **style.css** — файл стилей (наш «рецепт красоты»)
- 

## ШАГ 2: Создаём основу страницы (index.html)

Откройте ваш любимый редактор (помните, мы советовали VS Code?) и создайте новый файл. Вставьте в него этот код:

```
html
<!DOCTYPE html>
<html lang="ru">
<head>
  <meta charset="UTF-8">
```

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Практикум №5: Особые стили</title>
<!-- Подключаем наш файл стилей -->
<link rel="stylesheet" href="style.css">

<!-- Подключаем красивый шрифт с Google Fonts -->
<link rel="preconnect" href="https://fonts.googleapis.com">
<link rel="preconnect" href="https://fonts.gstatic.com" crossorigin>
<link href="https://fonts.googleapis.com/css2?
family=Playfair+Display:ital,wght@0,400;0,700;1,400&family=Roboto:wght@300;400;700&display=swap"
rel="stylesheet">
</head>
<body>
    <div class="container">
        <h1>Практикум №5: Создаём особые стили</h1>

        <p class="intro">Дорогой друг! Сегодня мы создадим два прекрасных стиля, которые помогут выделить самое важное и красиво завершить любое письмо.</p>

        <!-- МЕСТО ДЛЯ ОСОБО ВАЖНОГО ОБЪЯВЛЕНИЯ -->
        <div class="announcement-block">
            <h2>1. Особо важное объявление</h2>

            <div class="announcement" id="importantAnnouncement">
                <h3>ВНИМАНИЕ! Собрание садоводов</h3>
                <p>Дорогие друзья-садоводы! В эту субботу, 15 июня, в 10:00 у центральной беседки состоится важное собрание.</p>
                <p><strong>Тема:</strong> Подготовка к летнему фестивалю цветов.</p>
                <p><strong>Что взять:</strong> Идеи, хорошее настроение и рецепты пирогов для чаепития!</p>
                <p class="announcement-footer">Ждём всех обязательно! Не опаздывайте!</p>
            </div>
        </div>

        <div class="controls">
            <h4>Поэкспериментируйте с объявлением:</h4>
            <button onclick="changeAnnouncementColor()">Сменить цвет фона</button>
            <button onclick="changeAnnouncementFont()">Сменить шрифт заголовка</button>
            <button onclick="resetAnnouncement()">Вернуть как было</button>
        </div>
    </div>

    <!-- МЕСТО ДЛЯ ИЗЯЩНОЙ ПОДПИСИ -->
    <div class="signature-block">
```

## <h2>2. Изящная подпись</h2>

```
<div class="letter-example">
    <p class="letter-text">Мой дорогой внук,</p>
    <p class="letter-text">Пишу тебе это письмо, сидя у камина и вспоминая наши
летние прогулки по лесу. Как быстро летит время! Кажется, только вчера ты был маленьkim, а
теперь уже сам стал отцом.</p>
    <p class="letter-text">Хочу пожелать тебе и твоей семье здоровья, радости и
много солнечных дней. Помни, что семья — это самое главное богатство в жизни.</p>

    <!-- САМА ПОДПИСЬ -->
    <div class="signature" id="elegantSignature">
        С любовью,<br>
        <span class="signature-name">Твоя бабушка Анна</span>
    </div>
</div>

<div class="controls">
    <h4>Поэкспериментируйте с подписью:</h4>
    <button onclick="makeSignatureFancy()">Сделать ещё изящнее</button>
    <button onclick="changeSignatureColor()">Изменить цвет</button>
    <button onclick="resetSignature()">Вернуть как было</button>
</div>
</div>

<!-- ИНФОРМАЦИОННЫЙ БЛОК -->
<div class="info-box">
    <h3>Что мы сегодня узнали:</h3>
    <ul>
        <li>Как создать стиль для важных объявлений, которые нельзя пропустить</li>
        <li>Как оформить изящную подпись для писем и статей</li>
        <li>Как комбинировать несколько свойств CSS для достижения нужного
эффекта</li>
        <li>Как сделать стили интерактивными с помощью JavaScript</li>
    </ul>
    <p>Попробуйте изменить код в файле <strong>style.css</strong> и посмотрите, как
меняются стили!</p>
</div>
</div>

<script>
    // Функции для управления объявлением
    function changeAnnouncementColor() {
        const colors = ['#FFF0F5', '#FOFFF0', '#F0F8FF', '#FFFACD', '#FAF0E6'];

```

```

        const randomColor = colors[Math.floor(Math.random() * colors.length)];
        document.getElementById('importantAnnouncement').style.backgroundColor =
randomColor;
    }

    function changeAnnouncementFont() {
        const fonts = ["Playfair Display", serif, "Georgia", serif, "Times New Roman", serif,
'Arial, sans-serif'];
        const announcement = document.getElementById('importantAnnouncement');
        const currentFont = announcement.style.fontFamily || 'inherit';
        const newFont = fonts.find(font => font !== currentFont) || fonts[0];
        announcement.style.fontFamily = newFont;
    }

    function resetAnnouncement() {
        document.getElementById('importantAnnouncement').style = "";
    }

    // Функции для управления подписью
    function makeSignatureFancy() {
        const signature = document.getElementById('elegantSignature');
        signature.style.textShadow = '1px 1px 3px rgba(0,0,0,0.2)';
        signature.style.letterSpacing = '1px';
        signature.style.transform = 'skew(-5deg)';
        setTimeout(() => { signature.style.transform = 'skew(0deg)'; }, 1000);
    }

    function changeSignatureColor() {
        const colors = ['#8B4513', '#A0522D', '#D2691E', '#CD853F', '#8B7355'];
        const randomColor = colors[Math.floor(Math.random() * colors.length)];
        document.getElementById('elegantSignature').style.color = randomColor;
    }

    function resetSignature() {
        document.getElementById('elegantSignature').style = "";
    }
</script>
</body>
</html>
```

Сохраните этот файл как **index.html** в папке **Практикум5\_СпецСтили**.

---

## ШАГ 3: Создаём волшебные стили (style.css)

Теперь создайте новый файл и назовите его **style.css**. Вставьте в него этот код:

css

```
/*
ПРАКТИКУМ №5: Специальные стили
Стиль для "особо важного объявления" и "изящной подписи"
*/

/* ОСНОВНЫЕ НАСТРОЙКИ СТРАНИЦЫ */
body {
    font-family: 'Roboto', sans-serif;
    line-height: 1.6;
    color: #333;
    background-color: #f9f5f0; /* Нежный кремовый фон */
    padding: 20px;
    margin: 0;
}

.container {
    max-width: 800px;
    margin: 0 auto;
    background-color: white;
    padding: 30px;
    border-radius: 15px;
    box-shadow: 0 10px 30px rgba(0,0,0,0.1);
}

h1 {
    font-family: 'Playfair Display', serif;
    text-align: center;
    color: #8B4513; /* Тёплый коричневый */
    font-size: 2.5rem;
    margin-bottom: 30px;
    border-bottom: 2px solid #D2B48C;
    padding-bottom: 15px;
}

h2 {
    font-family: 'Playfair Display', serif;
    color: #A0522D; /* Терракотовый */
    margin-top: 40px;
}
```

```
padding-left: 15px;
border-left: 5px solid #D2B48C;
}

.intro {
    font-size: 1.1rem;
    color: #666;
    text-align: center;
    margin-bottom: 40px;
    font-style: italic;
}

/* БЛОКИ С ПРИМЕРАМИ */

.announcement-block, .signature-block {
    margin: 40px 0;
    padding: 25px;
    border-radius: 12px;
    background-color: #FAF8F5;
}

/*
СТИЛЬ 1: ОСОБО ВАЖНОЕ ОБЪЯВЛЕНИЕ
(Чтобы его нельзя было пропустить!)
*/
.announcement {
    background-color: #FFF0F5; /* Очень светлый розовый */
    border: 3px solid #FF69B4; /* Ярко-розовая рамка */
    border-radius: 10px;
    padding: 25px;
    margin: 20px 0;
    box-shadow: 0 5px 15px rgba(255,105,180,0.2);
    position: relative;
    overflow: hidden;
}

/* Добавляем декоративный уголок */
.announcement::before {
    content: "ВАЖНО!";
    position: absolute;
    top: 10px;
    right: -30px;
    background-color: #FF69B4;
    color: white;
    padding: 5px 40px;
}
```

```
    transform: rotate(45deg);
    font-weight: bold;
    font-size: 0.9rem;
    box-shadow: 0 3px 5px rgba(0,0,0,0.2);
}
```

```
.announcement h3 {
    font-family: 'Playfair Display', serif;
    color: #C71585; /* Средне-розовый */
    font-size: 1.8rem;
    margin-top: 0;
    text-align: center;
    border-bottom: 2px dashed #FFB6C1;
    padding-bottom: 10px;
}
```

```
.announcement p {
    font-size: 1.1rem;
    line-height: 1.7;
    margin-bottom: 15px;
}
```

```
.announcement strong {
    color: #DB7093;
    font-weight: 700;
}
```

```
.announcement-footer {
    text-align: center;
    font-weight: bold;
    color: #FF1493;
    font-size: 1.2rem;
    margin-top: 20px;
    padding-top: 15px;
    border-top: 1px solid #FFB6C1;
}
```

```
/*
СТИЛЬ 2: ИЗЯЩНАЯ ПОДПИСЬ
(Для завершения писем и статей)
*/
```

```
.letter-example {
    background-color: #FFFFFF;
    border: 1px solid #E6D5C3;
```

```
border-radius: 8px;  
padding: 30px;  
margin: 20px 0;  
box-shadow: 0 3px 10px rgba(0,0,0,0.05);  
}
```

```
.letter-text {  
    font-size: 1.1rem;  
    line-height: 1.8;  
    margin-bottom: 20px;  
    color: #555;  
}
```

```
.signature {  
    font-family: 'Playfair Display', serif;  
    font-size: 1.4rem;  
    color: #8B4513; /* Тёплый коричневый */  
    text-align: right;  
    margin-top: 40px;  
    padding-top: 20px;  
    border-top: 1px solid #D2B48C;  
    line-height: 1.8;  
    font-style: italic;  
}
```

```
.signature-name {  
    font-size: 1.8rem;  
    font-weight: bold;  
    color: #A0522D;  
    display: block;  
    margin-top: 5px;  
    text-shadow: 1px 1px 2px rgba(0,0,0,0.1);  
}
```

```
/* Панель управления для экспериментов */  
.controls {  
    background-color: #F5F5F5;  
    padding: 20px;  
    border-radius: 8px;  
    margin-top: 25px;  
    text-align: center;  
}
```

```
.controls h4 {
```

```
margin-top: 0;
color: #666;
font-size: 1.1rem;
}

.controls button {
background-color: #8B4513;
color: white;
border: none;
padding: 10px 20px;
margin: 5px;
border-radius: 5px;
cursor: pointer;
font-size: 1rem;
transition: all 0.3s;
}

.controls button:hover {
background-color: #A0522D;
transform: translateY(-2px);
box-shadow: 0 4px 8px rgba(0,0,0,0.2);
}

/* Информационный блок в конце */

.info-box {
background-color: #E8F4F8;
border-radius: 10px;
padding: 25px;
margin-top: 40px;
border-left: 6px solid #3498DB;
}

.info-box h3 {
color: #2C3E50;
margin-top: 0;
}

.info-box ul {
padding-left: 20px;
}

.info-box li {
margin-bottom: 10px;
line-height: 1.5;
```

```
}

.info-box p {
    font-style: italic;
    color: #555;
    margin-top: 15px;
}
```

Сохраните этот файл как **style.css** в той же папке, где лежит **index.html**.

---

## ШАГ 4: Запускаем и экспериментируем!

Теперь найдите в папке файл **index.html** и дважды щёлкните по нему. Он откроется в вашем браузере, и вы увидите:

### 1. Особо важное объявление

- Розовый фон с яркой рамкой
- Декоративный уголок с надписью "ВАЖНО!"
- Чёткий заголовок
- Удобочитаемый текст
- Кнопки для экспериментов

### 2. Изящная подпись

- Элегантный курсив
- Красивый коричневый цвет
- Текст выровнен по правому краю
- Имя выделено особенно красиво
- Кнопки для экспериментов

**Поэкспериментируйте с кнопками!** Нажмайте их и смотрите, как меняются стили в реальном времени.

---

## ШАГ 5: Разбираем код — как это работает?

Давайте разберём самые интересные части кода:

### Для объявления:

css

```
/* Декоративный уголок */
.announcement::before {
    content: "ВАЖНО!";
    position: absolute;
    top: 10px;
    right: -30px;
    background-color: #FF69B4;
    color: white;
    padding: 5px 40px;
    transform: rotate(45deg); /* Поворачиваем на 45 градусов! */
}
```

Волшебство здесь в `transform: rotate(45deg)` — мы поворачиваем элемент на 45 градусов, создавая эффект уголка.

### Для подписи:

css

```
.signature-name {
    font-size: 1.8rem;
    font-weight: bold;
    color: #A0522D;
    display: block;
    margin-top: 5px;
    text-shadow: 1px 1px 2px rgba(0,0,0,0.1); /* Лёгкая тень для объёма */
}
```

`text-shadow` добавляет лёгкую тень, делая текст чуть объёмнее и изящнее.

---

## ШАГ 6: Ваше домашнее задание

Попробуйте изменить стили самостоятельно:

### Задание 1: Создайте «Срочное объявление»

Измените стиль объявления так, чтобы оно выглядело СРОЧНЫМ:

- Красный фон (#FFE4E1)
- Тёмно-красная рамка (#B22222)
- Уголок с текстом «СРОЧНО!»
- Жёлтый цвет для самого важного текста

### Задание 2: Создайте «Официальную подпись»

Сделайте подпись более официальной:

- Синий цвет вместо коричневого
- Шрифт без засечек (Arial или Helvetica)
- Уберите курсив
- Добавьте небольшую печать под подписью (используйте ::after)

### Задание 3: Добавьте третий стиль

Придумайте и создайте свой собственный стиль! Например:

- «Нежное напоминание» (пастельные тона, лёгкие тени)
  - «Поздравление» (золотые цвета, праздничный вид)
  - «Благодарность» (тёплые зелёные оттенки)
- 

## Итог урока:

Сегодня вы научились:

1. **Создавать привлекающие внимание объявления** с помощью цвета, рамок и декоративных элементов
2. **Оформлять изящные подписи**, которые завершают текст с достоинством
3. **Комбинировать несколько CSS-свойств** для достижения сложных эффектов
4. **Добавлять интерактивность** с помощью JavaScript
5. **Использовать псевдоэлементы** (::before, ::after) для декоративных деталей

**Самое главное:** Теперь у вас есть два готовых «рецепта красоты», которые вы можете использовать на своих страницах. Просто скопируйте классы `.announcement` и `.signature` в свои проекты — и ваши объявления и подписи сразу станут красивыми!

**Запомните:** CSS — это цифровое рукоделие. Каждое свойство — это новая ниточка, каждый цвет — новый оттенок. Комбинируйте их смело, экспериментируйте, и вы создадите настоящие шедевры!

Удачи в творчестве! В следующей главе мы научимся создавать навигационное меню — основу любого сайта.

## ● Глава 8: Простое меню навигации – гордость любого сайта.

### ■ Создаем список ссылок (<ul> и <li>).

Дорогой друг! Пришло время научиться создавать навигацию — тот самый список ссылок, который есть на каждом сайте. Это как создать ящик с инструментами для вашего сайта: каждый инструмент (ссылка) лежит на своём месте, и вы легко можете найти нужный. Давайте разберёмся, как из обычного списка сделать красивую панель навигации.

---

## ЧАСТЬ 1: Что такое <ul> и <li>? (Проще простого!)

### Аналогия из жизни:

Представьте, что вы пишете список покупок на листочке:

text

Молоко

Хлеб

Яйца

Сахар

В HTML такой список выглядит так:

html

```
<ul>
  <li>Молоко</li>
  <li>Хлеб</li>
  <li>Яйца</li>
  <li>Сахар</li>
</ul>
```

### Расшифровка:

- <ul> — Unordered List (неупорядоченный список) — это контейнер для всего списка
- <li> — List Item (элемент списка) — это каждый отдельный пункт

По умолчанию браузер показывает такой список с точками (маркерами) перед каждым пунктом:

text

- Молоко
  - Хлеб
  - Яйца
  - Сахар
- 

## ЧАСТЬ 2: Делаем список ссылок

Теперь давайте превратим обычный список в список ссылок. Каждый пункт будет вести на другую страницу или раздел сайта.

### Базовый пример:

html

```
<ul>
  <li><a href="index.html">Главная</a></li>
  <li><a href="about.html">Обо мне</a></li>
  <li><a href="hobby.html">Моё хобби</a></li>
  <li><a href="photo.html">Фотоальбом</a></li>
  <li><a href="contacts.html">Контакты</a></li>
</ul>
```

### Что мы сделали:

1. Внутри каждого `<li>` поместили ссылку `<a>`
  2. У каждой ссылки есть атрибут `href` — это адрес, куда она ведёт
  3. Текст между `<a>` и `</a>` — это то, что видит пользователь
- 

## ЧАСТЬ 3: Практикум — создаём навигацию для сайта садовода

Давайте создадим реальный пример! Создайте папку **Навигация** и в ней файл **index.html**:

html

```
<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Мой сад - Навигация</title>
    <style>
        /* Базовые стили для всей страницы */
        body {
            font-family: 'Georgia', serif;
            background-color: #f0fff0; /* Светло-зелёный фон */
            color: #333;
            padding: 30px;
            line-height: 1.6;
        }

        h1 {
            color: #2e8b57; /* Цвет морской волны */
            text-align: center;
            border-bottom: 3px solid #98fb98;
            padding-bottom: 15px;
            max-width: 800px;
            margin: 0 auto 40px;
        }

        .container {
            max-width: 800px;
            margin: 0 auto;
            background-color: white;
            padding: 40px;
            border-radius: 15px;
            box-shadow: 0 10px 30px rgba(0,0,0,0.1);
        }

        .section {
            margin-bottom: 50px;
            padding: 25px;
            border-radius: 10px;
            background-color: #f9f9f9;
        }

        .section h2 {
```

```
color: #8b4513; /* Коричневый */  
margin-top: 0;  
padding-bottom: 10px;  
border-bottom: 2px solid #d2b48c;  
}  
  
.explanation {  
background-color: #e8f4f8;  
padding: 20px;  
border-radius: 10px;  
margin: 20px 0;  
border-left: 5px solid #3498db;  
}  
  
.explanation h3 {  
color: #2c3e50;  
margin-top: 0;  
}  
  
.code-block {  
background-color: #2c3e50;  
color: white;  
padding: 20px;  
border-radius: 10px;  
font-family: 'Courier New', monospace;  
margin: 20px 0;  
overflow-x: auto;  
}  
  
.example {  
border: 2px solid #e9ecef;  
padding: 25px;  
border-radius: 10px;  
margin: 25px 0;  
background-color: white;  
}  
  
.example-title {  
font-weight: bold;  
color: #27ae60;  
margin-bottom: 15px;  
font-size: 1.2rem;  
}  
</style>
```

```
</head>
<body>
<div class="container">
    <h1>Создаём список ссылок: навигация для сайта</h1>

    <div class="explanation">
        <h3> Что мы будем делать:</h3>
        <p>Мы создадим 5 разных вариантов списка ссылок — от самого простого до красивого меню. Каждый следующий вариант будет всё красивее!</p>
    </div>
```

*<!-- ПРИМЕР 1: Самый простой список -->*

```
<div class="section">
    <h2>Пример 1: Простой список с маркерами</h2>

    <div class="code-block">
&lt;ul&gt;<br>
&nbs; &nbs;&lt;li&gt;&lt;a href="#home"&gt;Главная&lt;/a&gt;&lt;/li&gt;<br>
&nbs; &nbs;&lt;li&gt;&lt;a href="#garden"&gt;Мой сад&lt;/a&gt;&lt;/li&gt;<br>
&nbs; &nbs;&lt;li&gt;&lt;a href="#flowers"&gt;Цветы&lt;/a&gt;&lt;/li&gt;<br>
&nbs; &nbs;&lt;li&gt;&lt;a href="#vegetables"&gt;Овощи&lt;/a&gt;&lt;/li&gt;<br>
&nbs; &nbs;&lt;li&gt;&lt;a href="#contacts"&gt;Контакты&lt;/a&gt;&lt;/li&gt;<br>
&lt;/ul&gt;
    </div>
```

```
<div class="example">
    <div class="example-title">Как это выглядит:</div>
    <ul>
        <li><a href="#home">Главная</a></li>
        <li><a href="#garden">Мой сад</a></li>
        <li><a href="#flowers">Цветы</a></li>
        <li><a href="#vegetables">Овощи</a></li>
        <li><a href="#contacts">Контакты</a></li>
    </ul>
    <p style="color: #666; margin-top: 15px;"><em>Это обычный список с точками.</em></p>
```

Работает, но выглядит скучно.</em></p>

*<!-- ПРИМЕР 2: Без маркеров -->*

```
<div class="section">
    <h2>Пример 2: Убираем маркеры (точки)</h2>
```

```
<div class="code-block">
```

```
&lt;style&gt;<br>
 &nbsp;&nbsp;.list-no-bullets {<br>
 &nbsp;&nbsp;&nbsp;&nbsp;list-style-type: none; /* Убираем маркеры */<br>
 &nbsp;&nbsp;&nbsp;&nbsp;padding-left: 0; /* Убираем отступ слева */<br>
 &nbsp;&nbsp;}<br>
&lt;/style&gt;<br><br>
&lt;ul class="list-no-bullets"&gt;<br>
 &nbsp;&nbsp;&lt;li&gt;&lt;a href="#home"&gt;Главная&lt;/a&gt;&lt;/li&gt;<br>
 &nbsp;&nbsp;&lt;li&gt;&lt;a href="#garden"&gt;Мой сад&lt;/a&gt;&lt;/li&gt;<br>
 &nbsp;&nbsp;&lt;li&gt;&lt;a href="#flowers"&gt;Цветы&lt;/a&gt;&lt;/li&gt;<br>
 &nbsp;&nbsp;&lt;li&gt;&lt;a href="#vegetables"&gt;Овощи&lt;/a&gt;&lt;/li&gt;<br>
 &nbsp;&nbsp;&lt;li&gt;&lt;a href="#contacts"&gt;Контакты&lt;/a&gt;&lt;/li&gt;<br>
&lt;/ul&gt;
</div>
```

```
<div class="example">
  <div class="example-title">Как это выглядит:</div>
  <ul style="list-style-type: none; padding-left: 0;">
    <li><a href="#home">Главная</a></li>
    <li><a href="#garden">Мой сад</a></li>
    <li><a href="#flowers">Цветы</a></li>
    <li><a href="#vegetables">Овощи</a></li>
    <li><a href="#contacts">Контакты</a></li>
  </ul>
  <p style="color: #666; margin-top: 15px;"><em>Точек нет, но всё ещё
вертикальный список.</em></p>
</div>
</div>
```

```
<!-- ПРИМЕР 3: Горизонтальный список -->
<div class="section">
  <h2>Пример 3: Делаем горизонтальный список</h2>
```

```
<div class="code-block">
&lt;style&gt;<br>
 &nbsp;&nbsp;.horizontal-list {<br>
 &nbsp;&nbsp;&nbsp;&nbsp;list-style-type: none;<br>
 &nbsp;&nbsp;&nbsp;&nbsp;padding-left: 0;<br>
 &nbsp;&nbsp;&nbsp;&nbsp;display: flex; /* Волшебное свойство! */<br>
 &nbsp;&nbsp;}<br>
 &nbsp;&nbsp;<br>
 &nbsp;&nbsp;.horizontal-list li {<br>
 &nbsp;&nbsp;&nbsp;&nbsp;margin-right: 20px; /* Отступ между пунктами */<br>
 &nbsp;&nbsp;&nbsp;}<br>
```

```

</style><br><br>
<ul class="horizontal-list"><br>
    &nbsp;&nbsp;&lt;li&gt;&lt;a href="#home"&gt;Главная&lt;/a&gt;&lt;/li&gt;<br>
    &nbsp;&nbsp;&lt;li&gt;&lt;a href="#garden"&gt;Мой сад&lt;/a&gt;&lt;/li&gt;<br>
    &nbsp;&nbsp;&lt;li&gt;&lt;a href="#flowers"&gt;Цветы&lt;/a&gt;&lt;/li&gt;<br>
    &nbsp;&nbsp;&lt;li&gt;&lt;a href="#vegetables"&gt;Овощи&lt;/a&gt;&lt;/li&gt;<br>
    &nbsp;&nbsp;&lt;li&gt;&lt;a href="#contacts"&gt;Контакты&lt;/a&gt;&lt;/li&gt;<br>
</ul>
</div>

<div class="example">
    <div class="example-title">Как это выглядит:</div>
    <ul style="list-style-type: none; padding-left: 0; display: flex;">
        <li style="margin-right: 20px;"><a href="#home">Главная</a></li>
        <li style="margin-right: 20px;"><a href="#garden">Мой сад</a></li>
        <li style="margin-right: 20px;"><a href="#flowers">Цветы</a></li>
        <li style="margin-right: 20px;"><a href="#vegetables">Овощи</a></li>
        <li><a href="#contacts">Контакты</a></li>
    </ul>
    <p style="color: #666; margin-top: 15px;"><em>Теперь ссылки в одну строку! Уже
похоже на меню.</em></p>
</div>
</div>

<!-- ПРИМЕР 4: Красивые кнопки -->
<div class="section">
    <h2>Пример 4: Превращаем в красивые кнопки</h2>

    <div class="code-block">
<style><br>
.button-list {<br>
    &nbsp;&nbsp;&nbsp;&nbsp;list-style-type: none;<br>
    &nbsp;&nbsp;&nbsp;&nbsp;padding-left: 0;<br>
    &nbsp;&nbsp;&nbsp;&nbsp;display: flex;<br>
    &nbsp;&nbsp;&nbsp;&nbsp;background-color: #2e8b57; /* Зелёный фон */<br>
    &nbsp;&nbsp;&nbsp;&nbsp;border-radius: 10px;<br>
    &nbsp;&nbsp;&nbsp;&nbsp;padding: 10px;<br>
    &nbsp;&nbsp;}<br>
    &nbsp;&nbsp;<br>
.button-list li {<br>
    &nbsp;&nbsp;&nbsp;&nbsp;margin-right: 5px;<br>
    &nbsp;&nbsp;}<br>
    &nbsp;&nbsp;<br>
.button-list a {<br>

```

```
&nbsp;&nbsp;&nbsp;&nbsp;color: white;<br>
&nbsp;&nbsp;&nbsp;&nbsp;text-decoration: none; /* Убираем подчёркивание */<br>
&nbsp;&nbsp;&nbsp;&nbsp;padding: 12px 24px;<br>
&nbsp;&nbsp;&nbsp;&nbsp;display: block;<br>
&nbsp;&nbsp;&nbsp;&nbsp;border-radius: 5px;<br>
&nbsp;&nbsp;&nbsp;&nbsp;transition: background-color 0.3s;<br>
&nbsp;&nbsp;}<br>
&nbsp;&nbsp;<br>
&nbsp;&nbsp;.button-list a:<br>
&nbsp;&nbsp;&nbsp;&nbsp;background-color: #3cb371; /* Светлее при наведении */<br>
&nbsp;&nbsp;}<br>
</style>
</div>
```

```
<div class="example">
    <div class="example-title">Как это выглядит:</div>
    <ul style="list-style-type: none; padding-left: 0; display: flex; background-color: #2e8b57; border-radius: 10px; padding: 10px;">
        <li style="margin-right: 5px;"><a href="#home" style="color: white; text-decoration: none; padding: 12px 24px; display: block; border-radius: 5px; background-color: rgba(255,255,255,0.1);">Главная</a></li>
        <li style="margin-right: 5px;"><a href="#garden" style="color: white; text-decoration: none; padding: 12px 24px; display: block; border-radius: 5px; background-color: rgba(255,255,255,0.1);">Мой сад</a></li>
        <li style="margin-right: 5px;"><a href="#flowers" style="color: white; text-decoration: none; padding: 12px 24px; display: block; border-radius: 5px; background-color: rgba(255,255,255,0.1);">Цветы</a></li>
        <li style="margin-right: 5px;"><a href="#vegetables" style="color: white; text-decoration: none; padding: 12px 24px; display: block; border-radius: 5px; background-color: rgba(255,255,255,0.1);">Овощи</a></li>
        <li><a href="#contacts" style="color: white; text-decoration: none; padding: 12px 24px; display: block; border-radius: 5px; background-color: rgba(255,255,255,0.1);">Контакты</a></li>
    </ul>
    <p style="color: #666; margin-top: 15px;"><em>Настоящее красивое меню!</em></p>
```

Попробуйте навести курсор на кнопки.</em></p>

```
</div>
</div>
```

```
<!-- ПРИМЕР 5: Полноценная навигация -->
```

```
<div class="section">
    <h2>Пример 5: Полноценная навигация с иконками</h2>
```

```
<div class="code-block">
```

```
&lt;style&gt;<br>
  /* Стили для всей навигации */<br>
  .garden-nav {<br>
    list-style-type: none;<br>
    padding-left: 0;<br>
    display: flex;<br>
    background: linear-gradient(to right, #2e8b57, #3cb371);<br>
    border-radius: 15px;<br>
    padding: 0;<br>
    overflow: hidden;<br>
    box-shadow: 0 5px 15px rgba(0,0,0,0.2);<br>
  }<br>
  /* Каждый пункт меню */<br>
  .garden-nav li {<br>
    flex: 1; /* Все пункты одинаковой ширины */<br>
    text-align: center;<br>
    border-right: 1px solid rgba(255,255,255,0.2);<br>
  }<br>
  .garden-nav li:last-child {<br>
    border-right: none;<br>
  }<br>
  /* Ссылки */<br>
  .garden-nav a {<br>
    color: white;<br>
    text-decoration: none;<br>
    padding: 20px 10px;<br>
    display: block;<br>
    font-weight: bold;<br>
    font-size: 1.1rem;<br>
    transition: all 0.3s;<br>
    position: relative;<br>
  }<br>
  /* Иконки перед текстом */<br>
  .garden-nav a::before {<br>
    content: "";<br>
    display: inline-block;<br>
    width: 24px;<br>
    height: 24px;<br>
    margin-right: 10px;<br>
    vertical-align: middle;<br>
```

```

        &nbsp;&nbsp;&nbsp;background-size: contain;<br>
        &nbsp;&nbsp;&nbsp;background-repeat: no-repeat;<br>
        &nbsp;&nbsp;&nbsp;background-position: center;<br>
        &nbsp;&nbsp;}<br>
&nbsp;&nbsp;<br>
        &nbsp;&nbsp; /* Разные иконки для разных пунктов */<br>
&nbsp;&nbsp;.home-link::before { background-image: url('data:image/svg+xml;utf8,<svg
xmlns="http://www.w3.org/2000/svg" viewBox="0 0 24 24" fill="white"><path d="M10 20v-6h4v6h5v-
8h3L12 3 2 12h3v8z"/></svg>'); }<br>
&nbsp;&nbsp;.garden-link::before { background-image: url('data:image/svg+xml;utf8,<svg
xmlns="http://www.w3.org/2000/svg" viewBox="0 0 24 24" fill="white"><path d="M12 2C8.13 2 5 5.13 5
9c0 5.25 7 13 7 13s7-7.75 7-13c0-3.87-3.13-7-7zm0 9.5c-1.38 0-2.5-1.12-2.5-2.5s1.12-2.5 2.5-2.5 2.5 1.12
2.5 2.5-1.12 2.5-2.5 2.5z"/></svg>'); }<br>
&nbsp;&nbsp;.flowers-link::before { background-image: url('data:image/svg+xml;utf8,<svg
xmlns="http://www.w3.org/2000/svg" viewBox="0 0 24 24" fill="white"><path d="M12 4.5C7 4.5 2.73 7.61
1 12c1.73 4.39 6 7.5 11 7.5s9.27-3.11 11-7.5c-1.73-4.39-6-7.5-11-7.5zM12 17c-2.76 0-5-2.24-5-5s2.24-5 5-5
5 2.24 5-2.24 5-5 5zm0-8c-1.66 0-3 1.34-3 3s1.34 3 3 3-1.34 3-3-1.34-3-3-3z"/></svg>'); }<br>
&nbsp;&nbsp;<br>
        &nbsp;&nbsp; /* Эффект при наведении */<br>
&nbsp;&nbsp;.garden-nav a:hover {<br>
&nbsp;&nbsp;&nbsp;&nbsp;background-color: rgba(255,255,255,0.2);<br>
&nbsp;&nbsp;&nbsp;&nbsp;transform: translateY(-2px);<br>
&nbsp;&nbsp;}<br>
&nbsp;&nbsp;<br>
        &nbsp;&nbsp; /* Активный пункт */<br>
&nbsp;&nbsp;.garden-nav a.active {<br>
&nbsp;&nbsp;&nbsp;&nbsp;background-color: rgba(255,255,255,0.3);<br>
&nbsp;&nbsp;&nbsp;&nbsp;box-shadow: inset 0 0 10px rgba(0,0,0,0.2);<br>
&nbsp;&nbsp;}<br>
</style><br><br>
<ul class="garden-nav"><br>
&nbsp;&nbsp;<li><a href="#home" class="home-link active">Главная</a></li><br>
&nbsp;&nbsp;<li><a href="#garden" class="garden-link">Мой сад</a></li><br>
&nbsp;&nbsp;<li><a href="#flowers" class="flowers-link">Цветы</a></li><br>
&nbsp;&nbsp;<li><a href="#vegetables">Овощи</a></li><br>
&nbsp;&nbsp;<li><a href="#contacts">Контакты</a></li><br>
</ul>
</div>

<div class="example">
    <div class="example-title">Как это выглядит:</div>
    <style>
        .garden-nav-final {
            list-style-type: none;

```

```
padding-left: 0;
display: flex;
background: linear-gradient(to right, #2e8b57, #3cb371);
border-radius: 15px;
padding: 0;
overflow: hidden;
box-shadow: 0 5px 15px rgba(0,0,0,0.2);
margin: 0;
}

.garden-nav-final li {
    flex: 1;
    text-align: center;
    border-right: 1px solid rgba(255,255,255,0.2);
}
.garden-nav-final li:last-child {
    border-right: none;
}

.garden-nav-final a {
    color: white;
    text-decoration: none;
    padding: 20px 10px;
    display: block;
    font-weight: bold;
    font-size: 1.1rem;
    transition: all 0.3s;
    position: relative;
}
.garden-nav-final a:hover {
    background-color: rgba(255,255,255,0.2);
    transform: translateY(-2px);
}
.garden-nav-final a.active {
    background-color: rgba(255,255,255,0.3);
    box-shadow: inset 0 0 10px rgba(0,0,0,0.2);
}
/* Иконки */
.icon {
    display: inline-block;
```

```

        width: 24px;
        height: 24px;
        margin-right: 10px;
        vertical-align: middle;
        background-size: contain;
        background-repeat: no-repeat;
        background-position: center;
    }

.home-icon { background-image: url('data:image/svg+xml;utf8,<svg
xmlns="http://www.w3.org/2000/svg" viewBox="0 0 24 24" fill="white"><path d="M10 20v-6h4v6h5v-
8h3L12 3 2 12h3v8z"/></svg>'); }

.garden-icon { background-image: url('data:image/svg+xml;utf8,<svg
xmlns="http://www.w3.org/2000/svg" viewBox="0 0 24 24" fill="white"><path d="M12 2C8.13 2 5 5.13 5
9c0 5.25 7 13 7 13s7-7.75 7-13c0-3.87-3.13-7-7zm0 9.5c-1.38 0-2.5-1.12-2.5-2.5s1.12-2.5 2.5-2.5 2.5 1.12
2.5 2.5-1.12 2.5-2.5 2.5z"/></svg>'); }

.flowers-icon { background-image: url('data:image/svg+xml;utf8,<svg
xmlns="http://www.w3.org/2000/svg" viewBox="0 0 24 24" fill="white"><path d="M12 4.5C7 4.5 2.73 7.61
1 12c1.73 4.39 6 7.5 11 7.5s9.27-3.11 11-7.5c-1.73-4.39-6-7.5-11-7.5zM12 17c-2.76 0-5-2.24-5-5s2.24-5 5-5
5 2.24 5-5 5zm0-8c-1.66 0-3 1.34-3 3s1.34 3 3 3-1.34 3-3-1.34-3-3-3z"/></svg>'); }

.vegetables-icon { background-image: url('data:image/svg+xml;utf8,<svg
xmlns="http://www.w3.org/2000/svg" viewBox="0 0 24 24" fill="white"><path d="M12 2C6.48 2 2 6.48 2
12s4.48 10 10 10 10-4.48 10-10s17.52 2 12 2zm-2 15l-5-5 1.41-1.41l10 14.17l7.59-7.59l19 8l-9
9z"/></svg>'); }

.contacts-icon { background-image: url('data:image/svg+xml;utf8,<svg
xmlns="http://www.w3.org/2000/svg" viewBox="0 0 24 24" fill="white"><path d="M20 4H4c-1.1 0-1.99.9-
1.99 2L2 18c0 1.1.9 2 2 2h16c1.1 0 2-.9 2-2V6c0-1.1-.9-2-2zm0 14H4V8l8 5 8-5v10zm-8-7L4 6h16l-8
5z"/></svg>'); }

</style>

<ul class="garden-nav-final">
    <li><a href="#home" class="active"><span class="icon home-
icon"></span>Главная</a></li>
    <li><a href="#garden"><span class="icon garden-icon"></span>Мой
сад</a></li>
    <li><a href="#flowers"><span class="icon flowers-
icon"></span>Цветы</a></li>
    <li><a href="#vegetables"><span class="icon vegetables-
icon"></span>Овощи</a></li>
    <li><a href="#contacts"><span class="icon contacts-
icon"></span>Контакты</a></li>
</ul>

<div style="display: flex; justify-content: center; margin-top: 20px;">
```

```
<button onclick="changeActiveNav()" style="padding: 10px 20px; background-color: #3498db; color: white; border: none; border-radius: 5px; cursor: pointer; font-size: 1rem;">Поменять активный пункт</button>
</div>
```

```
<p style="color: #666; margin-top: 15px;"><em>Профессиональное меню с иконками! Наведите курсор и попробуйте кнопку выше.</em></p>
</div>
</div>
```

```
<div class="explanation">
  <h3>⌚ Что мы узнали:</h3>
  <ul>
    <li><strong>&lt;ul&gt;</strong> создаёт список, <strong>&lt;li&gt;</strong> — каждый пункт списка</li>
      <li>Чтобы сделать ссылку, помещаем <strong>&lt;a href="#"&gt;</strong> внутрь <strong>&lt;li&gt;</strong></li>
        <li><code>list-style-type: none</code> убирает маркеры (точки)</li>
        <li><code>display: flex</code> делает список горизонтальным</li>
        <li>Можно добавлять иконки, градиенты, тени, эффекты при наведении</li>
        <li>Класс <code>active</code> показывает, какой пункт сейчас активен</li>
      </ul>
    </div>
```

```
<div style="text-align: center; margin-top: 40px; padding: 20px; background-color: #fff8e1; border-radius: 10px; border: 2px dashed #ffb300;">
  <h3 style="color: #d35400;">Ваше домашнее задание:</h3>
  <p>Создайте своё меню для сайта о вашем хобби! Используйте цвета, которые нравятся вам.</p>
  <button onclick="showHomework()" style="padding: 12px 30px; background-color: #e74c3c; color: white; border: none; border-radius: 8px; cursor: pointer; font-size: 1.1rem; margin-top: 10px;">Показать подсказки для задания</button>
  <div id="homeworkTips" style="display: none; margin-top: 20px; text-align: left; padding: 15px; background-color: white; border-radius: 8px;">
    <p><strong>Идеи для меню:</strong></p>
    <ul>
      <li><strong>Для вязания:</strong> Главная, Мои работы, Схемы, Полезные советы, Магазин ниток</li>
      <li><strong>Для садовода:</strong> Главная, Мой сад, Цветник, Огород, Рецепты, Фотосезона</li>
      <li><strong>Для кулинара:</strong> Главная, Рецепты, Видеоуроки, Мои секреты, Гостевые</li>
    </ul>
    <p><strong>Цветовые схемы:</strong></p>
  </div>
```

```

<ul>
    <li>Вязание: нежно-розовый (#FFE4E1) и сиреневый (#E6E6FA)</li>
    <li>Садовод: зелёный (#2E8B57) и жёлтый (#FFD700)</li>
    <li>Кулинария: томатный (#FF6347) и золотой (#FFD700)</li>
</ul>
</div>
</div>

</div>

<script>
// Функция для смены активного пункта меню
function changeActiveNav() {
    const navItems = document.querySelectorAll('.garden-nav-final a');
    navItems.forEach(item => item.classList.remove('active'));

    // Выбираем случайный пункт (кроме текущего активного)
    const activeIndex = Math.floor(Math.random() * navItems.length);
    navItems[activeIndex].classList.add('active');
}

// Функция для показа домашнего задания
function showHomework() {
    const tips = document.getElementById('homeworkTips');
    if (tips.style.display === 'none') {
        tips.style.display = 'block';
    } else {
        tips.style.display = 'none';
    }
}
</script>
</body>
</html>

```

Сохраните этот файл и откройте его в браузере. Вы увидите 5 разных примеров списков ссылок, от самого простого до полноценного меню с иконками!

---

## ЧАСТЬ 4: Ключевые CSS-свойства для списков ссылок

### 1. Убираем маркеры:

css

```
ul {  
    list-style-type: none; /* Убирает точки/цифры */  
    padding-left: 0; /* Убирает отступ слева */  
}
```

### 2. Делаем горизонтальный список:

css

```
ul {  
    display: flex; /* Волшебное свойство! */  
}  
  
li {  
    margin-right: 20px; /* Отступ между пунктами */  
}
```

### 3. Стили для ссылок:

css

```
a {  
    text-decoration: none; /* Убираем подчёркивание */  
    color: inherit; /* Наследуем цвет от родителя */  
    padding: 10px 20px; /* Внутренние отступы */  
    display: block; /* Чтобы padding работал правильно */  
}
```

### 4. Эффект при наведении:

css

```
a:hover {  
    background-color: #f0f0f0; /* Меняем фон */  
    color: #ff0000; /* Меняем цвет текста */  
}
```

## 5. Активный пункт:

css

```
a.active {  
    background-color: #007bff; /* Синий фон */  
    color: white; /* Белый текст */  
}
```

---

## ЧАСТЬ 5: Готовые примеры для копирования

### Простое горизонтальное меню:

html

```
<style>  
.simple-menu {  
    list-style: none;  
    padding: 0;  
    display: flex;  
    gap: 15px;  
}  
.simple-menu a {  
    text-decoration: none;  
    color: #333;  
    padding: 8px 15px;  
    border-radius: 5px;  
}  
.simple-menu a:hover {  
    background-color: #f0f0f0;  
}  
</style>  
  
<ul class="simple-menu">  
    <li><a href="index.html">Главная</a></li>  
    <li><a href="about.html">Обо мне</a></li>  
    <li><a href="hobby.html">Хобби</a></li>  
</ul>
```

## Меню с иконками:

html

```
<style>
.icon-menu {
    list-style: none;
    padding: 0;
    display: flex;
    background: #4CAF50;
    border-radius: 10px;
    overflow: hidden;
}
.icon-menu li {
    flex: 1;
    text-align: center;
}
.icon-menu a {
    color: white;
    text-decoration: none;
    padding: 15px;
    display: block;
    font-weight: bold;
}
.icon-menu a:hover {
    background: rgba(255,255,255,0.2);
}
</style>

<ul class="icon-menu">
    <li><a href="#home"> Главная</a></li>
    <li><a href="#garden"> Сад</a></li>
    <li><a href="#photos"> Фото</a></li>
    <li><a href="#contacts"> Контакты</a></li>
</ul>
```

---

## ЧАСТЬ 6: Частые ошибки и как их исправить

### Ошибка 1: Ссылки слишком близко друг к другу

**Решение:** Добавьте margin или padding:

css

```
li {  
    margin-right: 20px; /* ИЛИ */  
    padding: 0 10px;  
}
```

### Ошибка 2: Подчёркивание у ссылок

**Решение:**

css

```
a {  
    text-decoration: none;  
}
```

### Ошибка 3: Разные размеры у пунктов меню

**Решение:**

css

```
li {  
    flex: 1; /* Все пункты одинаковой ширины */  
    text-align: center;  
}
```

### Ошибка 4: Меню не помещается на маленьком экране

**Решение:**

css

```
@media (max-width: 768px) {
```

```
ul {  
    flex-direction: column; /* Вертикально на мобильных */  
}  
  
li {  
    margin-right: 0;  
    margin-bottom: 10px;  
}  
}
```

---

## ИТОГ:

Теперь вы умеете создавать списки ссылок любой сложности! От простого вертикального списка до профессионального меню с иконками и эффектами.

### Запомните главные шаги:

1. Создаём `<ul>` с `<li>` внутри
2. В каждый `<li>` добавляем ссылку `<a>`
3. Убираем маркеры: `list-style-type: none`
4. Делаем горизонтально: `display: flex`
5. Украшаем: добавляем цвета, отступы, эффекты

**Совет:** Начните с простого меню, а когда освоите его — добавляйте сложные эффекты. CSS позволяет создавать бесконечное количество вариантов дизайна!

Поздравляю! Теперь ваш сайт может иметь удобную навигацию, как у профессиональных сайтов. В следующей главе мы сделаем меню ещё лучше — добавим эффекты при наведении мыши!

- Превращаем скучный список в красивую горизонтальную панель с помощью CSS (убираем маркеры, делаем элементы в строку, добавляем фон и отступы).

Дорогой друг! Сегодня мы совершим настояще волшебство. Мы возьмём скучный вертикальный список с точками и превратим его в элегантную горизонтальную панель навигации. Это как взять старый комод, перекрасить его, поменять ручки и получить красивый предмет мебели!

---

## ЧАСТЬ 1: Исходный материал — "скучный список"

Давайте начнём с того, что обычно получается у начинающих:

html

```
<ul>
  <li><a href="#">Главная</a></li>
  <li><a href="#">Обо мне</a></li>
  <li><a href="#">Моё хобби</a></li>
  <li><a href="#">Фотогалерея</a></li>
  <li><a href="#">Контакты</a></li>
</ul>
```

Как это выглядит в браузере:

text

- Главная
- Обо мне
- Моё хобби
- Фотогалерея
- Контакты

Скучно, не правда ли? Давайте это исправим!

---

## ЧАСТЬ 2: Пошаговое превращение

Я создал для вас интерактивный пример, где вы сможете увидеть каждый шаг превращения. Создайте файл **панель.html** и скопируйте этот код:

```
html
<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Волшебство CSS: Из списка в панель</title>
    <style>
        /* Общие стили для страницы */
        * {
            box-sizing: border-box;
        }

        body {
            font-family: 'Georgia', serif;
            line-height: 1.6;
            color: #333;
            background-color: #f9f5f0;
            padding: 30px;
            max-width: 1000px;
            margin: 0 auto;
        }

        h1 {
            text-align: center;
            color: #8b4513;
            font-size: 2.5rem;
            margin-bottom: 40px;
            padding-bottom: 15px;
            border-bottom: 3px solid #d2b48c;
        }

        h2 {
            color: #a0522d;
            margin-top: 40px;
            padding-left: 15px;
            border-left: 5px solid #d2b48c;
        }
    

```

```
}

/* Контейнер для всех примеров */
.превращение-блок {
    background: white;
    padding: 30px;
    margin: 30px 0;
    border-radius: 15px;
    box-shadow: 0 10px 30px rgba(0,0,0,0.1);
}

/* Блок с кодом */
.код-блок {
    background: #2c3e50;
    color: #ecf0f1;
    padding: 25px;
    border-radius: 10px;
    font-family: 'Courier New', monospace;
    margin: 25px 0;
    overflow-x: auto;
    line-height: 1.8;
}

.css-свойство {
    color: #3498db; /* Синий для CSS свойств */
}

.css-значение {
    color: #2ecc71; /* Зелёный для значений */
}

.css-комментарий {
    color: #95a5a6; /* Серый для комментариев */
}

/* Блок с результатом */
.результат-блок {
    border: 3px dashed #e0e0e0;
    padding: 40px;
    border-radius: 10px;
    margin: 25px 0;
    background-color: #f8f9fa;
    min-height: 150px;
    display: flex;
```

```
    align-items: center;
    justify-content: center;
}

/* Панель управления */
.панель-управления {
    background: #34495e;
    color: white;
    padding: 25px;
    border-radius: 10px;
    margin: 30px 0;
}

.кнопка-шаг {
    background: #3498db;
    color: white;
    border: none;
    padding: 15px 25px;
    margin: 10px 5px;
    border-radius: 8px;
    cursor: pointer;
    font-size: 1.1rem;
    transition: all 0.3s;
}

.кнопка-шаг:hover {
    background: #2980b9;
    transform: translateY(-2px);
}

.активная-кнопка {
    background: #2ecc71 !important;
    box-shadow: 0 0 10px rgba(46, 204, 113, 0.5);
}

/* Информационные блоки */
.подсказка {
    background: #fff8e1;
    padding: 20px;
    border-radius: 10px;
    margin: 20px 0;
    border-left: 5px solid #f39c12;
}
```

```
.важно {
    background: #ffeaa7;
    padding: 20px;
    border-radius: 10px;
    margin: 20px 0;
    border-left: 5px solid #e74c3c;
    font-weight: bold;
}

/* Стили для списка на каждом этапе */
.исходный-список {
    font-size: 1.2rem;
    padding-left: 40px;
}

.этап1-без-маркеров {
    list-style-type: none;
    padding-left: 0;
    font-size: 1.2rem;
}

.этап2-горизонтальный {
    list-style-type: none;
    padding-left: 0;
    display: flex;
    font-size: 1.2rem;
}

.этап3-красивые-ссылки {
    list-style-type: none;
    padding-left: 0;
    display: flex;
    font-size: 1.2rem;
}

.этап3-красивые-ссылки a {
    text-decoration: none;
    color: #2c3e50;
    padding: 12px 25px;
    display: block;
    border-radius: 5px;
    transition: all 0.3s;
    border: 1px solid #ddd;
    margin: 0 5px;
}
```

```
}

.этап3-красивые-ссылки a:hover {
    background-color: #f8f9fa;
    transform: translateY(-2px);
    box-shadow: 0 4px 8px rgba(0,0,0,0.1);
}

.этап4-с-фоном {
    list-style-type: none;
    padding-left: 0;
    display: flex;
    font-size: 1.2rem;
    background: linear-gradient(135deg, #2c3e50, #4a6491);
    border-radius: 10px;
    padding: 10px;
}

.этап4-с-фоном a {
    text-decoration: none;
    color: white;
    padding: 15px 30px;
    display: block;
    border-radius: 5px;
    transition: all 0.3s;
    margin: 0 5px;
    text-align: center;
}

.этап4-с-фоном a:hover {
    background-color: rgba(255,255,255,0.2);
    transform: translateY(-2px);
}

.этап5-финальный {
    list-style-type: none;
    padding-left: 0;
    display: flex;
    font-size: 1.2rem;
    background: linear-gradient(135deg, #8b4513, #a0522d);
    border-radius: 15px;
    padding: 0;
    overflow: hidden;
    box-shadow: 0 10px 25px rgba(139, 69, 19, 0.3);
}
```

```
        }

.этап5-финальный li {
    flex: 1;
    text-align: center;
    border-right: 1px solid rgba(255,255,255,0.2);
}

.этап5-финальный li:last-child {
    border-right: none;
}

.этап5-финальный a {
    text-decoration: none;
    color: white;
    padding: 20px 15px;
    display: block;
    transition: all 0.3s;
    font-weight: bold;
    font-size: 1.1rem;
}

.этап5-финальный a:hover {
    background-color: rgba(255,255,255,0.15);
}

.этап5-финальный .активная-ссылка {
    background-color: rgba(255,255,255,0.25);
    box-shadow: inset 0 0 10px rgba(0,0,0,0.2);
}

/* Анимация перехода между этапами */
.результат-блок {
    transition: all 0.5s ease;
}

</style>
</head>
<body>
    <h1>Волшебство CSS: Из скучного списка в красивую панель</h1>

    <div class="подсказка">
        <h3 style="margin-top: 0; color: #d35400;">⊕ Что мы будем делать:</h3>
        <p>Мы пройдём 5 шагов превращения. На каждом шаге добавляем одно-два CSS-свойства. К концу урока у нас будет профессиональная панель навигации!</p>

```

```

</div>

<div class="превращение-блок">
    <h2>Исходный HTML-код (наш "сырой материал")</h2>

    <div class="код-блок">
        &lt;ul id="menu"&gt;<br>
        &nbsp;&nbsp;&lt;li&gt;&lt;a href="#"&gt;Главная&lt;/a&gt;&lt;/li&gt;<br>
        &nbsp;&nbsp;&lt;li&gt;&lt;a href="#"&gt;Обо мне&lt;/a&gt;&lt;/li&gt;<br>
        &nbsp;&nbsp;&lt;li&gt;&lt;a href="#"&gt;Моё хобби&lt;/a&gt;&lt;/li&gt;<br>
        &nbsp;&nbsp;&lt;li&gt;&lt;a href="#"&gt;Фотогалерея&lt;/a&gt;&lt;/li&gt;<br>
        &nbsp;&nbsp;&lt;li&gt;&lt;a href="#"&gt;Контакты&lt;/a&gt;&lt;/li&gt;<br>
        &lt;/ul&gt;
    </div>

    <div class="важно">
         <strong>Обратите внимание:</strong> У списка есть id="menu". Это как имя, чтобы мы могли обращаться к нему в CSS.
    </div>
</div>

<!-- ПАНЕЛЬ УПРАВЛЕНИЯ -->
<div class="панель-управления">
    <h2 style="color: white; margin-top: 0;">Управляйте превращением:</h2>
    <p style="color: #ecf0f1;">Нажмите на кнопки по порядку. Каждая кнопка добавляет новый CSS-стиль.</p>

    <div style="text-align: center; margin: 25px 0;">
        <button class="кнопка-шаг" id="шаг1" onclick="показатьЭтап(1)">Шаг 1: Убираем маркеры</button>
        <button class="кнопка-шаг" id="шаг2" onclick="показатьЭтап(2)">Шаг 2: Делаем горизонтально</button>
        <button class="кнопка-шаг" id="шаг3" onclick="показатьЭтап(3)">Шаг 3: Украшаем ссылки</button>
        <button class="кнопка-шаг" id="шаг4" onclick="показатьЭтап(4)">Шаг 4: Добавляем фон</button>
        <button class="кнопка-шаг" id="шаг5" onclick="показатьЭтап(5)">Шаг 5: Финальные штрихи</button>
        <button class="кнопка-шаг" onclick="показатьЭтап(0)" style="background: #e74c3c;">Сбросить</button>
    </div>
</div>

<!-- БЛОК ДЛЯ РЕЗУЛЬТАТА -->

```

```
<div class="превращение-блок">
  <h2>Результат: <span id="этап-название">Исходный список</span></h2>

  <div class="код-блок" id="css-код">
<span class="css-комментарий"/* Пока CSS-стили не добавлены */</span><br>
<span class="css-комментарий"/* Нажмите кнопки выше, чтобы добавить стили */</span>
  </div>

  <div class="результат-блок" id="результат">
    <ul class="исходный-список">
      <li><a href="#">Главная</a></li>
      <li><a href="#">Обо мне</a></li>
      <li><a href="#">Моё хобби</a></li>
      <li><a href="#">Фотогалерея</a></li>
      <li><a href="#">Контакты</a></li>
    </ul>
  </div>

  <div class="подсказка" id="объяснение">
    <h3 style="margin-top: 0; color: #27ae60;"> Объяснение текущего этапа:</h3>
    <p>Это исходный HTML-список. Браузер отображает его как вертикальный список с точками (маркерами) перед каждым пунктом.</p>
    <p>Чтобы начать превращение, нажмите "Шаг 1: Убираем маркеры" на панели выше.</p>
  </div>
</div>

<!-- ТЕОРЕТИЧЕСКАЯ ЧАСТЬ --&gt;
&lt;div class="превращение-блок"&gt;
  &lt;h2&gt;Подробное объяснение каждого шага&lt;/h2&gt;

  &lt;div id="подробности-шагов"&gt;
    &lt;div class="подсказка"&gt;
      &lt;h3 style="margin-top: 0; color: #3498db;"&gt;Выберите шаг на панели управления, чтобы увидеть подробное объяснение&lt;/h3&gt;
      &lt;p&gt;Здесь будет появляться детальное описание того, что делает каждый CSS-код и почему он важен.&lt;/p&gt;
    &lt;/div&gt;
  &lt;/div&gt;
&lt;/div&gt;

<!-- ГОТОВЫЕ ПРИМЕРЫ --&gt;
&lt;div class="превращение-блок"&gt;
  &lt;h2&gt;Готовые примеры для копирования&lt;/h2&gt;</pre>
```

```
<div class="подсказка">
    <h3 style="margin-top: 0; color: #9b59b6;">💡 Быстрые стили для ваших
проектов</h3>
    <p>Скопируйте любой из этих готовых стилей в ваш CSS-файл.</p>
</div>

<h3>Пример 1: Простая элегантная панель</h3>
<div class="код-блок">
<span class="css-комментарий">/* Простая горизонтальная панель */</span><br>
<span class="css-свойство">#menu</span> {<br>
    &nbsp;&nbsp;<span class="css-свойство">list-style-type</span>: <span class="css-
значение">none</span>;<br>
    &nbsp;&nbsp;<span class="css-свойство">padding</span>: <span class="css-значение">0</span>;<br>
    &nbsp;&nbsp;<span class="css-свойство">display</span>: <span class="css-значение">flex</span>;<br>
    &nbsp;&nbsp;<span class="css-свойство">gap</span>: <span class="css-значение">20px</span>;<br>
    &nbsp;&nbsp;<span class="css-свойство">background-color</span>: <span class="css-
значение">#f8f9fa</span>;<br>
    &nbsp;&nbsp;<span class="css-свойство">padding</span>: <span class="css-
значение">15px</span>;<br>
    &nbsp;&nbsp;<span class="css-свойство">border-radius</span>: <span class="css-
значение">10px</span>;<br>
}<br><br>
<span class="css-свойство">#menu a</span> {<br>
    &nbsp;&nbsp;<span class="css-свойство">text-decoration</span>: <span class="css-
значение">none</span>;<br>
    &nbsp;&nbsp;<span class="css-свойство">color</span>: <span class="css-значение">#333</span>;<br>
    &nbsp;&nbsp;<span class="css-свойство">padding</span>: <span class="css-значение">10px
20px</span>;<br>
    &nbsp;&nbsp;<span class="css-свойство">border-radius</span>: <span class="css-
значение">5px</span>;<br>
    &nbsp;&nbsp;<span class="css-свойство">transition</span>: <span class="css-значение">background-
color 0.3s</span>;<br>
}<br><br>
<span class="css-свойство">#menu a:hover</span> {<br>
    &nbsp;&nbsp;<span class="css-свойство">background-color</span>: <span class="css-
значение">#e9ecf</span>;<br>
}
</div>

<h3>Пример 2: Современная тёмная панель</h3>
<div class="код-блок">
<span class="css-комментарий">/* Современная тёмная панель */</span><br>
<span class="css-свойство">#menu</span> {<br>
```

```
&nbsp;&nbsp;<span class="css-свойство">list-style-type</span>: <span class="css-значение">none</span>;<br>
&nbsp;&nbsp;<span class="css-свойство">padding</span>: <span class="css-значение">0</span>;<br>
&nbsp;&nbsp;<span class="css-свойство">display</span>: <span class="css-значение">flex</span>;<br>
&nbsp;&nbsp;<span class="css-свойство">background</span>: <span class="css-значение">linear-gradient(135deg, #2c3e50, #34495e)</span>;<br>
&nbsp;&nbsp;<span class="css-свойство">border-radius</span>: <span class="css-значение">12px</span>;<br>
&nbsp;&nbsp;<span class="css-свойство">overflow</span>: <span class="css-значение">hidden</span>;<br>
&nbsp;&nbsp;<span class="css-свойство">box-shadow</span>: <span class="css-значение">0 8px 20px rgba(0,0,0,0.2)</span>;<br>
}<br><br>
<span class="css-свойство">#menu li</span> {<br>
&nbsp;&nbsp;<span class="css-свойство">flex</span>: <span class="css-значение">1</span>;<br>
&nbsp;&nbsp;<span class="css-свойство">text-align</span>: <span class="css-значение">center</span>;<br>
}<br><br>
<span class="css-свойство">#menu a</span> {<br>
&nbsp;&nbsp;<span class="css-свойство">text-decoration</span>: <span class="css-значение">none</span>;<br>
&nbsp;&nbsp;<span class="css-свойство">color</span>: <span class="css-значение">white</span>;<br>
&nbsp;&nbsp;<span class="css-свойство">padding</span>: <span class="css-значение">18px 15px</span>;<br>
&nbsp;&nbsp;<span class="css-свойство">display</span>: <span class="css-значение">block</span>;<br>
&nbsp;&nbsp;<span class="css-свойство">font-weight</span>: <span class="css-значение">bold</span>;<br>
&nbsp;&nbsp;<span class="css-свойство">transition</span>: <span class="css-значение">all 0.3s</span>;<br>
}<br><br>
<span class="css-свойство">#menu a:hover</span> {<br>
&nbsp;&nbsp;<span class="css-свойство">background-color</span>: <span class="css-значение">rgba(255,255,255,0.1)</span>;<br>
&nbsp;&nbsp;<span class="css-свойство">transform</span>: <span class="css-значение">translateY(-2px)</span>;<br>
}<br>
</div>
```

<div class="важно">



<strong>Совет:</strong> Скопируйте любой из этих примеров, подставьте свой

id вместо #menu и наслаждайтесь готовой панелью навигации!

</div>

</div>

```
<!-- ДОМАШНЕЕ ЗАДАНИЕ -->
<div style="background: linear-gradient(135deg, #8b4513, #a0522d); color: white; padding: 30px; border-radius: 15px; margin: 40px 0;">
    <h2 style="color: white; margin-top: 0;">Домашнее задание</h2>

    <div style="display: grid; grid-template-columns: repeat(auto-fit, minmax(250px, 1fr)); gap: 20px; margin-top: 20px;">
        <div style="background: rgba(255,255,255,0.1); padding: 20px; border-radius: 10px; backdrop-filter: blur(10px);">
            <h3 style="color: white;">Задание 1: Панель для сайта о вязании</h3>
            <p>Создайте панель в нежно-розовых тонах:</p>
            <ul style="color: #f0f0f0;">
                <li>Фон: #FFE4E1 (лавандовый румянец)</li>
                <li>Цвет ссылок: #8B4513 (коричневый)</li>
                <li>Эффект при наведении: #FFDAB9 (персиковый пух)</li>
                <li>Добавьте иконки:   </li>
            </ul>
        </div>

        <div style="background: rgba(255,255,255,0.1); padding: 20px; border-radius: 10px; backdrop-filter: blur(10px);">
            <h3 style="color: white;">Задание 2: Панель для садовода</h3>
            <p>Создайте "зелёную" панель:</p>
            <ul style="color: #f0f0f0;">
                <li>Градиент: от #2E8B57 до #3CB371</li>
                <li>Цвет ссылок: белый</li>
                <li>Добавьте разделители между пунктами</li>
                <li>Сделайте активный пункт с тенью внутри</li>
            </ul>
        </div>

        <div style="background: rgba(255,255,255,0.1); padding: 20px; border-radius: 10px; backdrop-filter: blur(10px);">
            <h3 style="color: white;">Задание 3: Эксперимент!</h3>
            <p>Придумайте свою панель:</p>
            <ul style="color: #f0f0f0;">
                <li>Используйте необычный градиент</li>
                <li>Добавьте анимацию при наведении</li>
                <li>Сделайте скруглённые углы разного радиуса</li>
                <li>Добавьте тень с размытием</li>
            </ul>
        </div>
    </div>
```

```
<div style="text-align: center; margin-top: 25px;">
    <button onclick="показатьРешение()" style="padding: 15px 30px; background: white; color: #a0522d; border: none; border-radius: 8px; cursor: pointer; font-size: 1.1rem; font-weight: bold; transition: all 0.3s;">
        Показать решение для задания 1
    </button>
</div>

<div id="решение" style="display: none; margin-top: 25px; padding: 20px; background: rgba(255,255,255,0.9); border-radius: 10px; color: #333;">
    <h3 style="color: #8b4513;">Решение для задания 1:</h3>
    <div class="код-блок" style="background: #fff; color: #333; border: 1px solid #ddd;">
<span class="css-комментарий"/* Панель для сайта о вязании */</span><br>
<span class="css-свойство">#вязка-меню</span> {<br>
    &nbsp;&nbsp;<span class="css-свойство">list-style-type</span>: <span class="css-значение">none</span>;<br>
    &nbsp;&nbsp;<span class="css-свойство">padding</span>: <span class="css-значение">0</span>;<br>
    &nbsp;&nbsp;<span class="css-свойство">display</span>: <span class="css-значение">flex</span>;<br>
    &nbsp;&nbsp;<span class="css-свойство">background-color</span>: <span class="css-значение">#FFE4E1</span>;<br>
    &nbsp;&nbsp;<span class="css-свойство">border-radius</span>: <span class="css-значение">20px</span>;<br>
    &nbsp;&nbsp;<span class="css-свойство">padding</span>: <span class="css-значение">15px</span>;<br>
    &nbsp;&nbsp;<span class="css-свойство">box-shadow</span>: <span class="css-значение">0 5px 15px rgba(255, 182, 193, 0.4)</span>;<br>
}<br><br>
<span class="css-свойство">#вязка-меню а</span> {<br>
    &nbsp;&nbsp;<span class="css-свойство">text-decoration</span>: <span class="css-значение">none</span>;<br>
    &nbsp;&nbsp;<span class="css-свойство">color</span>: <span class="css-значение">#8B4513</span>;<br>
    &nbsp;&nbsp;<span class="css-свойство">padding</span>: <span class="css-значение">12px 25px</span>;<br>
    &nbsp;&nbsp;<span class="css-свойство">margin</span>: <span class="css-значение">0 5px</span>;<br>
    &nbsp;&nbsp;<span class="css-свойство">border-radius</span>: <span class="css-значение">50px</span>;<br>
    &nbsp;&nbsp;<span class="css-свойство">transition</span>: <span class="css-значение">all 0.3s</span>;<br>
    &nbsp;&nbsp;<span class="css-свойство">font-weight</span>: <span class="css-значение">bold</span>;<br>
}<br><br>
```

```

<span class="css-свойство">#вязка-меню a:hover {<br>
    &nbsp;&nbsp;<span class="css-свойство">background-color</span>: <span class="css-
значение">#FFDAB9;</span>;<br>
    &nbsp;&nbsp;<span class="css-свойство">transform</span>: <span class="css-
значение">scale(1.05)</span>;<br>
    &nbsp;&nbsp;<span class="css-свойство">box-shadow</span>: <span class="css-значение">0 3px 10px
rgba(139, 69, 19, 0.2)</span>;<br>
}>
    </div>
</div>
</div>

<script>
let текущийЭтап = 0;

// CSS-коды для каждого этапа
const cssКоды = [
    /* Исходный код без стилей */,
    /* Шаг 1: Убираем маркеры */#menu {\n    list-style-type: none;\n    padding-left: 0;\n},
    /* Шаг 2: Делаем горизонтально */#menu {\n    list-style-type: none;\n    padding-left: 0;\n    display: flex;\n},
    /* Шаг 3: Украшаем ссылки */#menu a {\n    list-style-type: none;\n    padding-left: 0;\n    display: flex;\n    background: linear-gradient(135deg, #2c3e50, #4a6491);\n    border-radius: 10px;\n    padding: 10px;\n    color: white;\n    padding: 15px 30px;\n    display: block;\n    border-radius: 5px;\n    transition: all 0.3s;\n    margin: 0 5px;\n    text-align: center;\n}\n#menu a:hover {\n    background-color: #f8f9fa;\n    transform: translateY(-2px);\n    box-shadow: 0 4px 8px rgba(0,0,0,0.1);\n},
    /* Шаг 4: Добавляем фон */#menu {\n    list-style-type: none;\n    padding-left: 0;\n    display: flex;\n    background: linear-gradient(135deg, #2c3e50, #4a6491);\n    border-radius: 10px;\n    padding: 10px;\n    color: white;\n    padding: 15px 30px;\n    display: block;\n    border-radius: 5px;\n    transition: all 0.3s;\n    margin: 0 5px;\n    text-align: center;\n}\n#menu a:hover {\n    background-color: rgba(255,255,255,0.2);\n    transform: translateY(-2px);\n},
    /* Шаг 5: Финальные штрихи */#menu {\n    list-style-type: none;\n    padding-left: 0;\n    display: flex;\n    background: linear-gradient(135deg, #8b4513, #a0522d);\n    border-radius: 15px;\n    padding: 0;\n    overflow: hidden;\n    box-shadow: 0 10px 25px rgba(139, 69, 0.3);\n}\n#menu li {\n    flex: 1;\n    text-align: center;\n    border-right: 1px solid rgba(255,255,255,0.2);\n}\n#menu li:last-child {\n    border-right: none;\n}\n#menu a {\n    text-decoration: none;\n    color: white;\n    padding: 20px 15px;\n    display: block;\n    transition: all 0.3s;\n    font-weight: bold;\n    font-size: 1.1rem;\n}\n#menu a:hover {\n    background-color: rgba(255,255,255,0.15);\n}\n#menu .активная-ссылка {\n    background-color: rgba(255,255,255,0.25);\n    box-shadow: inset 0 0 10px rgba(0,0,0,0.2);\n}
];

// Объяснения для каждого этапа
const объяснения = [

```

‘Это исходный HTML-список. Браузер отображает его как вертикальный список с точками (маркерами) перед каждым пунктом.’,

‘**list-style-type: none**’ — убирает маркеры (точки) у списка.  
‘**padding-left: 0**’ — убирает отступ слева, который браузер добавляет

по умолчанию.

‘*Теперь у нас чистый список без лишних элементов!*’,

‘**display: flex**’ — это волшебное свойство! Оно превращает наш вертикальный список в горизонтальный.

Flexbox (гибкая коробка) — это современный способ расположения элементов. Он делает все элементы списка (**<li>**) в одну строку.

‘*Теперь наши пункты идут в строку, как в настоящем меню!*’,

‘Теперь украшаем сами ссылки:

‘**text-decoration: none**’ — убираем подчёркивание у ссылок.

‘**padding: 12px 25px**’ — добавляем внутренние отступы, чтобы ссылки были больше и удобнее для клика.

‘**display: block**’ — делаем ссылку блочным элементом, чтобы padding работал правильно.

‘**border-radius: 5px**’ — скругляем углы.

‘**transition: all 0.3s**’ — добавляем плавную анимацию при наведении.

Эффект при наведении (**<a>:hover**):

‘**transform: translateY(-2px)**’ — слегка поднимаем ссылку при наведении.

‘**box-shadow**’ — добавляем тень для эффекта “поднятия”.

‘*Теперь ссылки выглядят как красивые кнопки!*’,

‘Добавляем фон всему меню:

‘**background: linear-gradient(135deg, #2c3e50, #4a6491)**’ — добавляем градиентный фон (плавный переход от одного цвета к другому).

‘**border-radius: 10px**’ — скругляем углы у всего меню.

‘**padding: 10px**’ — добавляем внутренние отступы для всего меню.

Меняем цвет ссылок на белый, чтобы они контрастировали с тёмным фоном.

‘*Теперь у нас цельная красивая панель, а не просто набор ссылок!*’,

‘Финальные штрихи для профессионального вида:

‘**overflow: hidden**’ — обрезаем всё, что выходит за границы меню.

‘**box-shadow**’ — добавляем тень для объёмного эффекта.

```
<strong>flex: 1</strong> — делаем все пункты меню одинаковой ширины.<br>
<strong>border-right</strong> — добавляем вертикальные разделители между
пунктами.<br>
    <strong>:last-child</strong> — убираем разделитель у последнего пункта.<br>
    <strong>.активная-ссылка</strong> — специальный стиль для активной
страницы.<br>
<br>
<em>Вот и всё! Мы превратили скучный список в профессиональную панель
навигации!</em>`  
];  
  
// Названия этапов  
const названияЭтапов = [  
    "Исходный список",  
    "Без маркеров",  
    "Горизонтальное расположение",  
    "Красивые ссылки",  
    "С фоном",  
    "Финальная панель"  
];  
  
// Классы для каждого этапа  
const классыЭтапов = [  
    "исходный-список",  
    "этап1-без-маркеров",  
    "этап2-горизонтальный",  
    "этап3-красивые-ссылки",  
    "этап4-с-фоном",  
    "этап5-финальный"  
];  
  
function показатьЭтап(номер) {  
    текущийЭтап = номер;  
  
    // Обновляем результат  
    const результат = document.getElementById('результат');  
    результат.innerHTML = `  
        <ul class="${классыЭтапов[номер]}>  
            <li><a href="#" ${номер} = 5 ? 'class="активная-ссылка"' : ''>Главная</a></li>  
            <li><a href="#">Обо мне</a></li>  
            <li><a href="#">Моё хобби</a></li>  
            <li><a href="#">Фотогалерея</a></li>  
            <li><a href="#">Контакты</a></li>  
        </ul>
```

```

`;

// Обновляем CSS-код
document.getElementById('css-код').innerHTML = cssКоды[номер];

// Обновляем название этапа
document.getElementById('этап-название').textContent = названияЭтапов[номер];

// Обновляем объяснение
document.getElementById('объяснение').innerHTML = `
<h3 style="margin-top: 0; color: #27ae60;"> Объяснение этапа ${номер}: ${названияЭтапов[номер]}</h3>
<p>${объяснения[номер]}</p>
`;

// Обновляем кнопки
for (let i = 1; i <= 5; i++) {
    const кнопка = document.getElementById('шаг' + i);
    if (i <= номер) {
        кнопка.classList.add('активная-кнопка');
    } else {
        кнопка.classList.remove('активная-кнопка');
    }
}

// Обновляем подробности
let подробностиHTML = '';
for (let i = 1; i <= 5; i++) {
    подробностиHTML += `
<div style="margin-bottom: 25px; padding: 20px; background-color: ${i <= номер ? '#e8f6f3' : '#f5f5f5'}; border-radius: 8px; border-left: 5px solid ${i <= номер ? '#27ae60' : '#bdc3c7'};">
    <h3 style="margin-top: 0; color: ${i <= номер ? '#27ae60' : '#7f8c8d'};">Шаг ${i}: ${названияЭтапов[i]}</h3>
    <p>${i <= номер ? ' Выполнено' : ' Ещё не сделано'}</p>
    <p style="color: #666;">${объяснения[i]}</p>
</div>
`;
}
document.getElementById('подробности-шагов').innerHTML = подробностиHTML;
}

function показатьРешение() {
    const решение = document.getElementById('решение');

```

```
if (решение.style.display === 'none') {
    решение.style.display = 'block';
    setTimeout(() => {
        решение.scrollIntoView({ behavior: 'smooth' });
    }, 100);
} else {
    решение.style.display = 'none';
}
}

// Инициализация
показатьЭтап(0);

</script>
</body>
</html>
```

---

## ЧАСТЬ 3: Краткая шпаргалка по основным свойствам

### 1. Убираем маркеры:

css

```
ul {
    list-style-type: none; /* Убираем точки/цифры */
    padding-left: 0;      /* Убираем левый отступ */
}
```

### 2. Делаем горизонтально:

css

```
ul {
    display: flex;      /* Волшебное свойство! */
}
```

### 3. Распределяем равномерно:

css

```
li {
    flex: 1;           /* Все пункты одинаковой ширины */
```

```
    text-align: center; /* Текст по центру */
}
```

## 4. Стили для ссылок:

css

```
a {
    text-decoration: none; /* Без подчёркивания */
    color: white; /* Белый текст */
    padding: 15px 20px; /* Отступы внутри */
    display: block; /* Чтобы padding работал */
    transition: all 0.3s; /* Плавные анимации */
}
```

## 5. Эффект при наведении:

css

```
a:hover {
    background-color: rgba(255,255,255,0.2); /* Полупрозрачный фон */
    transform: translateY(-2px); /* Поднимаем на 2px */
}
```

## 6. Красивый фон:

css

```
ul {
    background: linear-gradient(135deg, #цвет1, #цвет2); /* Градиент */
    border-radius: 15px; /* Закруглённые углы */
    box-shadow: 0 10px 25px rgba(0,0,0,0.2); /* Тень */
}
```

---

## ИТОГ:

Теперь вы знаете весь секрет превращения скучного списка в красивую панель!  
Всего 5 простых шагов:

1. Убираем маркеры — `list-style-type: none`
2. Делаем горизонтально — `display: flex`
3. Украшаем ссылки — `padding, border-radius, transition`

4. **Добавляем фон** — background, box-shadow
5. **Финальные штрихи** — flex: 1, разделители, активное состояние

**Совет:** Начните с простого и добавляйте сложности постепенно. Сначала сделайте просто горизонтальный список, потом добавьте фон, потом — эффекты при наведении.

Поздравляю! Теперь вы можете создавать профессиональные панели навигации для своих сайтов. В следующей главе мы сделаем эти панели ещё умнее — добавим эффекты при наведении мыши!

## ● Глава 9: Чудеса при наведении мышки.

### ■ Псевдокласс :hover – что это такое?

Дорогой друг! Приготовьтесь к настоящему волшебству! Сегодня мы научимся оживлять наши веб-страницы. Представьте, что ваши ссылки и кнопки могут "чувствовать" приближение мышки и реагировать на неё: менять цвет, подниматься, дрожать от радости! Всё это возможно благодаря псевдоклассу :hover.

---

## ЧАСТЬ 1: Что такое :hover? (Проще простого!)

### Простая аналогия:

Представьте, что у вас есть кошка. Когда вы её просто видите — она спокойно лежит. Но когда вы **подносите руку, чтобы погладить** — она начинает мурлыкать, выгибает спинку, её шёрстка становится пушистее.

Элемент веб-страницы с :hover — это как такая кошка! Когда пользователь **наводит на него курсор мыши** — элемент "оживает" и показывает свою реакцию.

### Техническое определение:

:hover — это **псевдокласс CSS**, который применяет стили к элементу, **когда курсор мыши находится над ним**.

css

```
/* Базовый синтаксис */
элемент:hover {
    свойство: значение;
}
```

### Пример:

css

```
/* Обычная ссылка */
a {
    color: blue;
    text-decoration: none;
}

/* Та же ссылка при наведении */
a:hover {
    color: red;
    text-decoration: underline;
}
```

---

## ЧАСТЬ 2: Интерактивный учебник — увидеть магию своими глазами!

Давайте создадим файл **hover-магия.html** и посмотрим на волшебство в действии:

html

```
<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Волшебство :hover - Оживляем страницы!</title>
    <style>
        /* ОСНОВНЫЕ СТИЛИ СТРАНИЦЫ */
        * {
            box-sizing: border-box;
        }

        body {
            font-family: 'Georgia', serif;
            line-height: 1.6;
            color: #333;
            background: linear-gradient(135deg, #f5f7fa 0%, #c3cfe2 100%);
            padding: 20px;
            margin: 0;
            min-height: 100vh;
        }
    
```

```
}

.контейнер {
    max-width: 1200px;
    margin: 0 auto;
    background: white;
    border-radius: 20px;
    box-shadow: 0 20px 60px rgba(0,0,0,0.1);
    overflow: hidden;
}

.заголовок {
    background: linear-gradient(135deg, #8b4513, #a0522d);
    color: white;
    padding: 40px;
    text-align: center;
    position: relative;
}

.заголовок h1 {
    margin: 0;
    font-size: 2.8rem;
    text-shadow: 2px 2px 4px rgba(0,0,0,0.3);
}

.заголовок p {
    font-size: 1.2rem;
    opacity: 0.9;
    max-width: 800px;
    margin: 20px auto 0;
}

/* АНИМИРОВАННЫЙ ПОДЗАГОЛОВОК */

.магический-подзаголовок {
    text-align: center;
    padding: 30px;
    background: #fff8e1;
    border-bottom: 3px dashed #ffb300;
    font-size: 1.4rem;
    color: #d35400;
    position: relative;
    overflow: hidden;
}
```

```
.магический-подзаголовок:hover {
    animation: pulsate 1s infinite;
}

@keyframes pulsate {
    0% { background-color: #fff8e1; }
    50% { background-color: #ffecb3; }
    100% { background-color: #fff8e1; }
}

/* СЕТКА ДЛЯ ПРИМЕРОВ */
.лаборатория {
    display: grid;
    grid-template-columns: repeat(auto-fit, minmax(350px, 1fr));
    gap: 30px;
    padding: 40px;
}

.эксперимент-блок {
    background: #f8f9fa;
    border-radius: 15px;
    padding: 30px;
    box-shadow: 0 10px 30px rgba(0,0,0,0.08);
    border: 2px solid #e9ecef;
    transition: transform 0.3s;
}

.эксперимент-блок:hover {
    transform: translateY(-10px);
    border-color: #3498db;
    box-shadow: 0 15px 40px rgba(52, 152, 219, 0.2);
}

.номер-эксперимента {
    display: inline-block;
    background: #3498db;
    color: white;
    width: 36px;
    height: 36px;
    border-radius: 50%;
    text-align: center;
    line-height: 36px;
    font-weight: bold;
    margin-bottom: 20px;
}
```

```
        font-size: 1.2rem;
    }



### 


```

```
flex-direction: column;
gap: 20px;
}

/* ЭКСПЕРИМЕНТ 1: Простая ссылка */
.простая-ссылка {
    color: #2980b9;
    text-decoration: none;
    font-size: 1.3rem;
    padding: 10px 20px;
    border-radius: 5px;
    transition: all 0.3s;
}

.простая-ссылка:hover {
    color: #e74c3c;
    text-decoration: underline;
    background-color: #f8f9fa;
    transform: scale(1.05);
}

/* ЭКСПЕРИМЕНТ 2: Волшебная кнопка */
.волшебная-кнопка {
    background: linear-gradient(135deg, #2ecc71, #27ae60);
    color: white;
    border: none;
    padding: 15px 40px;
    font-size: 1.2rem;
    border-radius: 50px;
    cursor: pointer;
    font-weight: bold;
    transition: all 0.3s;
    box-shadow: 0 5px 15px rgba(46, 204, 113, 0.4);
    position: relative;
    overflow: hidden;
}

.волшебная-кнопка:hover {
    transform: translateY(-5px) scale(1.05);
    box-shadow: 0 10px 25px rgba(46, 204, 113, 0.6);
    background: linear-gradient(135deg, #27ae60, #219653);
}

.волшебная-кнопка:active {
```

```
        transform: translateY(0) scale(0.98);
    }

/* ЭФФЕКТ ВСПЫШКИ ПРИ НАВЕДЕНИИ */
.волшебная-кнопка::after {
    content: " ";
    position: absolute;
    top: 50%;
    left: 50%;
    width: 5px;
    height: 5px;
    background: rgba(255, 255, 255, 0.5);
    opacity: 0;
    border-radius: 100%;
    transform: scale(1, 1) translate(-50%);
    transform-origin: 50% 50%;
}

.волшебная-кнопка:hover::after {
    animation: ripple 1s ease-out;
}

@keyframes ripple {
    0% {
        transform: scale(0, 0);
        opacity: 0.5;
    }
    100% {
        transform: scale(40, 40);
        opacity: 0;
    }
}

/* ЭКСПЕРИМЕНТ 3: Карточка с поднятием */
.карточка {
    background: white;
    border-radius: 15px;
    padding: 25px;
    width: 250px;
    box-shadow: 0 5px 15px rgba(0,0,0,0.1);
    transition: all 0.4s cubic-bezier(0.175, 0.885, 0.32, 1.275);
}

.карточка:hover {
```

```
        transform: translateY(-20px) rotate(2deg);
        box-shadow: 0 20px 40px rgba(0,0,0,0.2);
    }

    .карточка img {
        width: 100%;
        border-radius: 10px;
        margin-bottom: 15px;
    }

    .карточка h4 {
        margin: 0 0 10px 0;
        color: #2c3e50;
    }

    .карточка p {
        color: #7f8c8d;
        margin: 0;
    }

/* ЭКСПЕРИМЕНТ 4: Меню с подсветкой */

.меню-подсветка {
    list-style: none;
    padding: 0;
    margin: 0;
    display: flex;
    background: #34495e;
    border-radius: 10px;
    overflow: hidden;
}

.меню-подсветка li {
    flex: 1;
}

.меню-подсветка a {
    color: white;
    text-decoration: none;
    padding: 20px;
    display: block;
    text-align: center;
    font-weight: bold;
    position: relative;
    transition: all 0.3s;
}
```

```
}

.меню-подсветка a:hover {
    background: #2c3e50;
}

/* Подсветка снизу */
.меню-подсветка a::after {
    content: "";
    position: absolute;
    bottom: 0;
    left: 50%;
    width: 0;
    height: 4px;
    background: #3498db;
    transition: all 0.3s;
    transform: translateX(-50%);
}

.меню-подсветка a:hover::after {
    width: 100%;
}

/* ЭКСПЕРИМЕНТ 5: Изображение с эффектом */
.изображение-эффект {
    width: 200px;
    height: 200px;
    border-radius: 10px;
    overflow: hidden;
    position: relative;
    cursor: pointer;
}

.изображение-эффект img {
    width: 100%;
    height: 100%;
    object-fit: cover;
    transition: all 0.5s;
}

.изображение-эффект:hover img {
    transform: scale(1.1);
    filter: brightness(0.8) saturate(1.2);
}
```

```
.изображение-эффект .подпись {
    position: absolute;
    bottom: -50px;
    left: 0;
    right: 0;
    background: rgba(0, 0, 0, 0.7);
    color: white;
    padding: 15px;
    transition: all 0.5s;
}

.изображение-эффект:hover .подпись {
    bottom: 0;
}

/* ЭКСПЕРИМЕНТ 6: Секретная панель */
.секретная-панель {
    background: #ecf0f1;
    border-radius: 10px;
    padding: 20px;
    position: relative;
    overflow: hidden;
}

.секретная-панель .секретное-сообщение {
    position: absolute;
    top: 0;
    left: 0;
    right: 0;
    bottom: 0;
    background: #9b59b6;
    color: white;
    display: flex;
    align-items: center;
    justify-content: center;
    font-size: 1.5rem;
    font-weight: bold;
    transform: translateY(100%);
    transition: transform 0.5s;
    border-radius: 10px;
}

.секретная-панель:hover .секретное-сообщение {
```

```
        transform: translateY(0);
    }

/* ТЕОРЕТИЧЕСКАЯ ЧАСТЬ */

.теория {
    padding: 40px;
    background: #f8f9fa;
    border-top: 3px solid #3498db;
}

.инфо-блок {
    background: #e8f4fc;
    padding: 25px;
    border-radius: 10px;
    margin: 20px 0;
    border-left: 5px solid #3498db;
}

.предупреждение {
    background: #fffea7;
    padding: 25px;
    border-radius: 10px;
    margin: 20px 0;
    border-left: 5px solid #f39c12;
}

/* АККОРДЕОН С FAQ */

.faq {
    margin: 40px 0;
}

.faq-вопрос {
    background: #2c3e50;
    color: white;
    padding: 20px;
    cursor: pointer;
    border-radius: 8px;
    margin-bottom: 10px;
    transition: background 0.3s;
    position: relative;
}

.faq-вопрос:hover {
    background: #34495e;
}
```

```
}

.faq-вопрос::after {
    content: '+';
    position: absolute;
    right: 20px;
    font-size: 1.5rem;
    transition: transform 0.3s;
}

.faq-вопрос.active::after {
    content: '-';
    transform: rotate(180deg);
}

.faq-ответ {
    background: white;
    padding: 0;
    border-radius: 0 0 8px 8px;
    max-height: 0;
    overflow: hidden;
    transition: all 0.3s;
    border: 1px solid #ecf0f1;
    border-top: none;
}

.faq-ответ.active {
    padding: 20px;
    max-height: 500px;
}

/* ДОМАШНЕЕ ЗАДАНИЕ */

.домашнее-задание {
    background: linear-gradient(135deg, #8b4513, #a0522d);
    color: white;
    padding: 40px;
    text-align: center;
    margin-top: 40px;
}

.задания-сетка {
    display: grid;
    grid-template-columns: repeat(auto-fit, minmax(300px, 1fr));
    gap: 20px;
}
```

```
        margin-top: 30px;
    }

.zадание-карточка {
    background: rgba(255, 255, 255, 0.1);
    padding: 25px;
    border-radius: 10px;
    backdrop-filter: blur(10px);
    border: 1px solid rgba(255, 255, 255, 0.2);
}

.zадание-карточка h4 {
    margin-top: 0;
    color: #ffd700;
}

</style>
</head>
<body>
    <div class="контейнер">
        <!-- ЗАГОЛОВОК -->
        <div class="заголовок">
            <h1>✨ Волшебство :hover ✨</h1>
            <p>Наведите курсор мыши на элементы ниже и увидите магию CSS в действии!</p>
        </div>

        <!-- ПОДЗАГОЛОВОК С АНИМАЦИЕЙ -->
        <div class="магический-подзаголовок">
             Каждый элемент на этой странице "оживает" при наведении курсора!
        </div>

        <!-- ЛАБОРАТОРИЯ ЭКСПЕРИМЕНТОВ -->
        <div class="лаборатория">
            <!-- ЭКСПЕРИМЕНТ 1: Простая ссылка -->
            <div class="эксперимент-блок">
                <div class="номер-эксперимента">1</div>
                <h3>Простая ссылка с :hover</h3>

                <div class="код-блок">
<span class="css-селектор">a</span> {<br>
&nbsp;&nbsp;<span class="css-свойство">color</span>: <span class="css-значение">#2980b9</span>;<br>
&nbsp;&nbsp;<span class="css-свойство">text-decoration</span>: <span class="css-значение">none</span>;<br>
} <br><br>
```

```
<span class="css-селектор">a:hover</span> {<br>
  &nbsp;&nbsp;<span class="css-свойство">color</span>: <span class="css-
  значение">#e74c3c</span>;<br>
  &nbsp;&nbsp;<span class="css-свойство">text-decoration</span>: <span class="css-
  значение">underline</span>;<br>
  &nbsp;&nbsp;<span class="css-свойство">background-color</span>: <span class="css-
  значение">#f8f9fa</span>;<br>
  &nbsp;&nbsp;<span class="css-свойство">transform</span>: <span class="css-
  значение">scale(1.05)</span>;<br>
}</div>

<div class="демо-область">
  <a href="#" class="простая-ссылка">Наведите на меня курсор!</a>
</div>

<div class="инфо-блок">
  <strong>Как это работает:</strong> Когда курсор оказывается над ссылкой,
  срабатывает псевдокласс <code>:hover</code> и применяются новые стили.
</div>
</div>

<!-- ЭКСПЕРИМЕНТ 2: Волшебная кнопка --&gt;
&lt;div class="эксперимент-блок"&gt;
  &lt;div class="номер-эксперимента"&gt;2&lt;/div&gt;
  &lt;h3&gt;Волшебная кнопка с эффектом&lt;/h3&gt;

  &lt;div class="код-блок"&gt;
&lt;span class="css-селектор"&gt;.кнопка&lt;/span&gt; {&lt;br&gt;
  &amp;nbsp;&amp;nbsp;&lt;span class="css-свойство"&gt;background&lt;/span&gt;: &lt;span class="css-значение"&gt;linear-
  gradient(...)&lt;/span&gt;;&lt;br&gt;
  &amp;nbsp;&amp;nbsp;&lt;span class="css-свойство"&gt;transition&lt;/span&gt;: &lt;span class="css-значение"&gt;all
  0.3s&lt;/span&gt;;&lt;br&gt;
}&lt;br&gt;&lt;br&gt;
&lt;span class="css-селектор"&gt;.кнопка:hover&lt;/span&gt; {&lt;br&gt;
  &amp;nbsp;&amp;nbsp;&lt;span class="css-свойство"&gt;transform&lt;/span&gt;: &lt;span class="css-значение"&gt;translateY(-
  5px) scale(1.05)&lt;/span&gt;;&lt;br&gt;
  &amp;nbsp;&amp;nbsp;&lt;span class="css-свойство"&gt;box-shadow&lt;/span&gt;: &lt;span class="css-значение"&gt;0 10px
  25px rgba(...)&lt;/span&gt;;&lt;br&gt;
}&lt;br&gt;&lt;br&gt;
&lt;span class="css-селектор"&gt;.кнопка:hover::after&lt;/span&gt; {&lt;br&gt;
  &amp;nbsp;&amp;nbsp;&lt;span class="css-свойство"&gt;animation&lt;/span&gt;: &lt;span class="css-значение"&gt;ripple 1s
  ease-out&lt;/span&gt;;&lt;br&gt;
}&lt;/div&gt;</pre>
```

```
</div>

<div class="демо-область">
    <button class="волшебная-кнопка">Нажми меня! (но сначала
наведи)</button>
</div>

<div class="инфо-блок">
    <strong>Эффект пульсации:</strong> Используем псевдоэлемент
<code>::after</code> вместе с <code>:hover</code> для создания волнового эффекта.
</div>
</div>

<!-- ЭКСПЕРИМЕНТ 3: Карточка с поднятием -->
<div class="эксперимент-блок">
    <div class="номер-эксперимента">3</div>
    <h3>Карточка, которая "поднимается"</h3>

    <div class="код-блок">
<span class="css-селектор">.карточка</span> {<br>
    &nbsp;&nbsp;<span class="css-свойство">transition</span>: <span class="css-значение">all 0.4s cubic-
bezier(...)</span>;<br>
    &nbsp;&nbsp;<span class="css-свойство">box-shadow</span>: <span class="css-значение">0 5px 15px
rgba(...)</span>;<br>
}<br><br>
<span class="css-селектор">.карточка:hover</span> {<br>
    &nbsp;&nbsp;<span class="css-свойство">transform</span>: <span class="css-значение">translateY(-
20px) rotate(2deg)</span>;<br>
    &nbsp;&nbsp;<span class="css-свойство">box-shadow</span>: <span class="css-значение">0 20px
40px rgba(...)</span>;<br>
}
    </div>

    <div class="демо-область">
        <div class="карточка">
            
                <h4>Волшебная карточка</h4>
                <p>Наведите на меня курсор, и я поднимусь!</p>
            </div>
        </div>
```

```
<div class="инфо-блок">
    <strong>cubic-bezier:</strong> Специальная функция для плавной
анимации. Делает движение более естественным.

</div>
</div>

<!-- ЭКСПЕРИМЕНТ 4: Меню с подсветкой -->
<div class="эксперимент-блок">
    <div class="номер-эксперимента">4</div>
    <h3>Меню с подсветкой снизу</h3>

    <div class="код-блок">
<span class="css-селектор">.меню a::after</span> {<br>
&nbsp;&nbsp;<span class="css-свойство">content</span>: <span class="css-значение">"</span>;<br>
&nbsp;&nbsp;<span class="css-свойство">position</span>: <span class="css-
значение">absolute</span>;<br>
&nbsp;&nbsp;<span class="css-свойство">bottom</span>: <span class="css-значение">0</span>;<br>
&nbsp;&nbsp;<span class="css-свойство">width</span>: <span class="css-значение">0</span>;<br>
}<br><br>
<span class="css-селектор">.меню a:hover::after</span> {<br>
&nbsp;&nbsp;<span class="css-свойство">width</span>: <span class="css-
значение">100%</span>;<br>
}

</div>

<div class="демо-область">
    <ul class="меню-подсветка">
        <li><a href="#">Главная</a></li>
        <li><a href="#">Обо мне</a></li>
        <li><a href="#">Портфолио</a></li>
        <li><a href="#">Контакты</a></li>
    </ul>
</div>

<div class="инфо-блок">
    <strong>Псевдоэлемент ::after:</strong> Создаёт невидимый элемент,
который становится видимым при наведении. Отличный способ добавить декоративные элементы
без изменения HTML.

</div>
</div>

<!-- ЭКСПЕРИМЕНТ 5: Изображение с эффектом -->
<div class="эксперимент-блок">
```

```

<div class="номер-эксперимента">5</div>
<h3>Изображение с выезжающей подписью</h3>

<div class="код-блок">
<span class="css-селектор">.изображение:hover img</span> {<br>
&nbsp;&nbsp;<span class="css-свойство">transform</span>: <span class="css-значение">scale(1.1)</span>;<br>
&nbsp;&nbsp;<span class="css-свойство">filter</span>: <span class="css-значение">brightness(0.8)</span>;<br>
}<br><br>
<span class="css-селектор">.изображение:hover .подпись</span> {<br>
&nbsp;&nbsp;<span class="css-свойство">bottom</span>: <span class="css-значение">0</span>;<br>
}
</div>

<div class="демо-область">
    <div class="изображение-эффект">
        
            <div class="подпись">Улыбающийся смайлик! 😊</div>
        </div>
    </div>

    <div class="инфо-блок">
        <strong>Несколько эффектов сразу:</strong> При наведении
одновременно увеличивается изображение, затемняется и появляется подпись.
        </div>
    </div>

<!-- ЭКСПЕРИМЕНТ 6: Секретная панель -->
<div class="эксперимент-блок">
    <div class="номер-эксперимента">6</div>
    <h3>Секретное сообщение</h3>

    <div class="код-блок">
<span class="css-селектор">.секретное-сообщение</span> {<br>
&nbsp;&nbsp;<span class="css-свойство">transform</span>: <span class="css-значение">translateY(100%)</span>;<br>
&nbsp;&nbsp;<span class="css-свойство">transition</span>: <span class="css-значение">transform
0.5s</span>;<br>
}<br><br>

```

```
<span class="css-селектор">.панель:hover .секретное-сообщение</span> {<br>
&nbsp;&nbsp;<span class="css-свойство">transform</span>: <span class="css-
значение">translateY(0)</span>;<br>
}<br>

</div>

<div class="демо-область">
  <div class="секретная-панель">
    <p>Наведите курсор на эту панель, чтобы увидеть секретное
сообщение!</p>
    <div class="секретное-сообщение">
      🎉 Поздравляю! Вы нашли секрет! 🎉
    </div>
  </div>
</div>

<div class="инфо-блок">
  <strong>Родительский селектор:</strong> Используем
<code>.родитель:hover .ребенок</code> чтобы анимировать дочерний элемент при наведении на
родителя.
</div>
</div>
</div>

<!-- ТЕОРЕТИЧЕСКАЯ ЧАСТЬ -->
<div class="теория">
  <h2 style="text-align: center; color: #2c3e50;">📚 Теория: Всё о :hover</h2>

  <div class="инфо-блок">
    <h3 style="color: #3498db;">Что такое псевдокласс?</h3>
    <p><strong>Псевдокласс</strong> — это специальное ключевое слово в CSS,
которое добавляется к селектору и определяет его <strong>особое состояние</strong>.</p>
    <p>Псевдоклассы начинаются с двоеточия: <code>:hover</code>,
<code>:active</code>, <code>:focus</code>, <code>:visited</code>.</p>
  </div>

  <div class="инфо-блок">
    <h3 style="color: #3498db;">Синтаксис :hover</h3>
    <div class="код-блок">
<span class="css-комментарий">/* Базовый синтаксис */</span><br>
<span class="css-селектор">селектор</span>:<code>:hover <br>
&nbsp;&nbsp;<span class="css-свойство">свойство</span>: <span class="css-
значение">значение</span>;<br>
}<br><br>
```

```
<span class="css-комментарий">/* Пример */</span><br>
<span class="css-селектор">button</span>:hover {<br>
  &nbsp;&nbsp;<span class="css-свойство">background-color</span>: <span class="css-значение">#e74c3c</span>;<br>
  &nbsp;&nbsp;<span class="css-свойство">color</span>: <span class="css-значение">white</span>;<br>
}<br>
  </div>
</div>

<!-- FAQ АККОРДЕОН -->
<div class="faq">
  <h3 style="color: #2c3e50; text-align: center;"> ? Часто задаваемые
  вопросы</h3>

  <div class="faq-вопрос" onclick="toggleFAQ(1)">Какие элементы
  поддерживают :hover?</div>
    <div class="faq-ответ" id="faq1">
      <p>Почти все HTML-элементы поддерживают <code>:hover</code>:</p>
      <ul>
        <li>Ссылки (<code>&lt;a&gt;</code>)</li>
        <li>Кнопки (<code>&lt;button&gt;</code>)</li>
        <li>Поля формы</li>
        <li>Изображения</li>
        <li>Любые блоки (<code>&lt;div&gt;</code>,
<code>&lt;p&gt;</code>)</li>
      </ul>
    </div>

    <div class="faq-вопрос" onclick="toggleFAQ(2)">Как сделать плавную анимацию?
  </div>
    <div class="faq-ответ" id="faq2">
      <p>Используйте свойство <code>transition</code>:</p>
      <div class="код-блок">
<span class="css-селектор">element</span> {<br>
  &nbsp;&nbsp;<span class="css-свойство">transition</span>: <span class="css-значение">all 0.3s
  ease</span>;<br>
}<br><br>
<span class="css-комментарий">/* all — все свойства */</span><br>
<span class="css-комментарий">/* 0.3s — длительность 0.3 секунды */</span><br>
<span class="css-комментарий">/* ease — тип анимации (плавный) */</span>
      </div>
    </div>
```

```
<div class="faq-вопрос" onclick="toggleFAQ(3)">Что будет на мобильных
устройствах?</div>
<div class="faq-ответ" id="faq3">
    <p>На мобильных устройствах <code>:hover</code> работает по-
разному:</p>
    <ul>
        <li><strong>Сенсорные экраны:</strong> Срабатывает при
касании</li>
        <li><strong>Проблема:</strong> После касания элемент остаётся в
состоянии :hover</li>
        <li><strong>Решение:</strong> Используйте <code>@media (hover:
hover)</code> для устройств с мышкой</li>
    </ul>
</div>

<div class="faq-вопрос" onclick="toggleFAQ(4)">Можно ли комбинировать
псевдоклассы?</div>
<div class="faq-ответ" id="faq4">
    <p>Да! Можно создавать сложные состояния:</p>
    <div class="код-блок">
<span class="css-комментарий">/* Ссылка, на которую навели И она активна */</span><br>
<span class="css-селектор">a:hover:active</span> {<br>
&nbsp;&nbsp;<span class="css-свойство">color</span>: <span class="css-
значение">#c0392b</span>;<br>
}<br><br>
<span class="css-комментарий">/* Последний элемент списка при наведении */</span><br>
<span class="css-селектор">li:last-child:hover</span> {<br>
&nbsp;&nbsp;<span class="css-свойство">background</span>: <span class="css-
значение">#ffeb3b</span>;<br>
}
</div>
</div>
</div>

<div class="предупреждение">
    <h3 style="color: #e74c3c;">⚠️ Важные правила использования :hover</h3>
    <ul>
        <li><strong>Не переусердствуйте!</strong> Слишком много анимаций
отвлекают</li>
        <li><strong>Сохраняйте контраст</strong> — текст должен оставаться
читаемым</li>
        <li><strong>Тестируйте на разных устройствах</strong> — особенно на
мобильных</li>
        <li><strong>Используйте transition</strong> для плавных переходов</li>
    </ul>
</div>
```

```
<li><strong>Проверяйте доступность</strong> — не все могут
использовать мышь</li>
</ul>
</div>

<div class="инфо-блок">
    <h3 style="color: #3498db;">Другие полезные псевдоклассы</h3>
    <div style="display: grid; grid-template-columns: repeat(auto-fit, minmax(250px,
1fr)); gap: 15px; margin-top: 20px;">
        <div style="background: white; padding: 15px; border-radius: 8px; border-left:
4px solid #3498db;">
            <strong>:active</strong><br>
            Когда элемент активирован (нажат)
        </div>
        <div style="background: white; padding: 15px; border-radius: 8px; border-left:
4px solid #2ecc71;">
            <strong>:focus</strong><br>
            Когда элемент в фокусе (выбран Tab)
        </div>
        <div style="background: white; padding: 15px; border-radius: 8px; border-left:
4px solid #e74c3c;">
            <strong>:visited</strong><br>
            Для уже посещённых ссылок
        </div>
        <div style="background: white; padding: 15px; border-radius: 8px; border-left:
4px solid #f39c12;">
            <strong>:nth-child()</strong><br>
            Для выбора определённых детей
        </div>
        </div>
    </div>
</div>

<!-- ДОМАШНЕЕ ЗАДАНИЕ --&gt;
&lt;div class="домашнее-задание"&gt;
    &lt;h2 style="color: white; margin-top: 0;"&gt;❶ Домашнее задание&lt;/h2&gt;
    &lt;p style="font-size: 1.2rem; max-width: 800px; margin: 0 auto;"&gt;Создайте свои
собственные :hover эффекты!&lt;/p&gt;

    &lt;div class="задания-сетка"&gt;
        &lt;div class="задание-карточка"&gt;
            &lt;h4&gt;Задание 1: "Дрожащая" кнопка&lt;/h4&gt;
            &lt;p&gt;Создайте кнопку, которая слегка дрожит при наведении.&lt;/p&gt;
            &lt;div class="код-блок" style="background: rgba(0,0,0,0.3);"&gt;</pre>
```

```
<strong>Подсказка:</strong><br>
transform: rotate(1deg);<br>
animation: shake 0.3s;

</div>
</div>

<div class="задание-карточка">
    <h4>Задание 2: Выпадающее меню</h4>
    <p>Сделайте меню, при наведении на которое выпадает подменю.</p>
    <div class="код-блок" style="background: rgba(0,0,0,0.3);">

<strong>Подсказка:</strong><br>
_submenu {<br>
&nbsp;&nbsp;display: none;<br>
}<br>
.menu:hover .submenu {<br>
&nbsp;&nbsp;display: block;<br>
}
    </div>
</div>

<div class="задание-карточка">
    <h4>Задание 3: Интерактивная галерея</h4>
    <p>Создайте галерею, где при наведении на фото остальные
затемняются.</p>
    <div class="код-блок" style="background: rgba(0,0,0,0.3);">

<strong>Подсказка:</strong><br>
.gallery:hover img:not(:hover) {<br>
&nbsp;&nbsp;filter: brightness(0.5);<br>
}
    </div>
</div>

<button onclick="показатьРешение()" style="margin-top: 30px; padding: 15px 40px;
background: white; color: #a0522d; border: none; border-radius: 50px; font-size: 1.1rem; font-weight:
bold; cursor: pointer; transition: all 0.3s;">
    Показать решение для задания 1
</button>

<div id="решение" style="display: none; margin-top: 30px; padding: 25px; background:
rgba(255,255,255,0.9); border-radius: 10px; color: #333;">
    <h3 style="color: #8b4513;">Решение для задания 1:</h3>
    <div class="код-блок" style="background: #fff; color: #333; border: 1px solid
#ddd;">
```

```
<span class="css-комментарий">/* Дрожащая кнопка */</span><br>
<span class="css-селектор">.дрожащая-кнопка</span> {<br>
  &nbsp;&nbsp;<span class="css-свойство">background</span>: <span class="css-значение">#3498db</span>;<br>
  &nbsp;&nbsp;<span class="css-свойство">color</span>: <span class="css-значение">white</span>;<br>
  &nbsp;&nbsp;<span class="css-свойство">padding</span>: <span class="css-значение">15px 30px</span>;<br>
  &nbsp;&nbsp;<span class="css-свойство">border</span>: <span class="css-значение">none</span>;<br>
  &nbsp;&nbsp;<span class="css-свойство">border-radius</span>: <span class="css-значение">8px</span>;<br>
  &nbsp;&nbsp;<span class="css-свойство">cursor</span>: <span class="css-значение">pointer</span>;<br>
  &nbsp;&nbsp;<span class="css-свойство">font-size</span>: <span class="css-значение">1.1rem</span>;<br>
  &nbsp;&nbsp;<span class="css-свойство">transition</span>: <span class="css-значение">all 0.3s</span>;<br>
}<br><br>
<span class="css-селектор">.дрожащая-кнопка:hover</span> {<br>
  &nbsp;&nbsp;<span class="css-свойство">animation</span>: <span class="css-значение">shake 0.5s infinite</span>;<br>
  &nbsp;&nbsp;<span class="css-свойство">background</span>: <span class="css-значение">#2980b9</span>;<br>
}<br><br>
<span class="css-селектор">@keyframes shake</span> {<br>
  &nbsp;&nbsp;0% { <span class="css-свойство">transform</span>: <span class="css-значение">rotate(0deg)</span>; }<br>
  &nbsp;&nbsp;25% { <span class="css-свойство">transform</span>: <span class="css-значение">rotate(1deg)</span>; }<br>
  &nbsp;&nbsp;50% { <span class="css-свойство">transform</span>: <span class="css-значение">rotate(0deg)</span>; }<br>
  &nbsp;&nbsp;75% { <span class="css-свойство">transform</span>: <span class="css-значение">rotate(-1deg)</span>; }<br>
  &nbsp;&nbsp;100% { <span class="css-свойство">transform</span>: <span class="css-значение">rotate(0deg)</span>; }<br>
}
</div>
</div>
</div>

<script>
  // Функция для FAQ аккордеона
  function toggleFAQ(num) {
```

```

const ответ = document.getElementById('faq' + num);
const вопрос = ответ.previousElementSibling;

if (ответ.classList.contains('active')) {
    ответ.classList.remove('active');
    вопрос.classList.remove('active');
} else {
    // Закрываем все остальные
    document.querySelectorAll('.faq-ответ.active').forEach(el => {
        el.classList.remove('active');
        el.previousElementSibling.classList.remove('active');
    });
}

// Открываем текущий
ответ.classList.add('active');
вопрос.classList.add('active');

}

// Функция для показа решения
function показатьРешение() {
    const решение = document.getElementById('решение');
    if (решение.style.display === 'none') {
        решение.style.display = 'block';
        setTimeout(() => {
            решение.scrollIntoView({ behavior: 'smooth' });
        }, 100);
    } else {
        решение.style.display = 'none';
    }
}

// Добавляем интерактивность для демо
document.addEventListener('DOMContentLoaded', function() {
    // Добавляем обработчик клика для волшебной кнопки
    document.querySelector('.волшебная-кнопка').addEventListener('click', function() {
        alert('🎉 Ура! Вы нажали на волшебную кнопку! 🎉');
    });
});

// Делаем ссылки в демо неактивными
document.querySelectorAll('.демо-область a').forEach(link => {
    link.addEventListener('click', function(e) {
        e.preventDefault();
        alert('Это демо-ссылка. В реальном сайте она бы куда-то вела!');
    });
});

```

```
        });
    });
}
</script>
</body>
</html>
```

---

## ЧАСТЬ 3: Ключевые моменты для запоминания

### 1. Базовый синтаксис:

css

```
селектор:hover {
    свойство: значение;
}
```

### 2. Всегда добавляйте transition для плавности:

css

```
элемент {
    transition: все 0.3s;
}
```

### 3. Самые популярные свойства для :hover:

#### Цвет и фон:

css

```
a:hover {
    color: #ff0000;      /* Меняем цвет текста */
    background-color: #f0f0f0; /* Меняем фон */
    background: linear-gradient(...); /* Градиентный фон */
}
```

#### Трансформации:

css

```
button:hover {
```

```
    transform: scale(1.1); /* Увеличиваем на 10% */  
    transform: translateY(-5px); /* Поднимаем на 5px */  
    transform: rotate(5deg); /* Поворачиваем на 5° */  
}
```

## Тени и границы:

css

```
div:hover {  
    box-shadow: 0 10px 20px rgba(0,0,0,0.2); /* Тень */  
    border-color: #3498db; /* Цвет рамки */  
    border-radius: 20px; /* Закругление */  
}
```

## Фильтры (для изображений):

css

```
img:hover {  
    filter: brightness(1.2); /* Делаем ярче */  
    filter: contrast(1.5); /* Увеличиваем контраст */  
    filter: grayscale(100%); /* Делаем чёрно-белым */  
}
```

---

# ЧАСТЬ 4: Готовые рецепты для копирования

## Рецепт 1: Простая кнопка с подъёмом

css

```
.кнопка {  
    background: #3498db;  
    color: white;  
    padding: 12px 24px;  
    border: none;  
    border-radius: 8px;  
    cursor: pointer;  
    transition: all 0.3s;  
    font-size: 1rem;  
}  
  
.кнопка:hover {
```

```
background: #2980b9;
transform: translateY(-3px);
box-shadow: 0 5px 15px rgba(41, 128, 185, 0.4);
}
```

## Рецепт 2: Ссылка с подчёркиванием снизу

css

```
.ссылка {
    color: #2c3e50;
    text-decoration: none;
    position: relative;
    padding-bottom: 5px;
}
```

```
.ссылка::after {
    content: "";
    position: absolute;
    bottom: 0;
    left: 0;
    width: 0;
    height: 2px;
    background: #e74c3c;
    transition: width 0.3s;
}
```

```
.ссылка:hover::after {
    width: 100%;
}
```

## Рецепт 3: Карточка с эффектом

css

```
.карточка {
    background: white;
    border-radius: 10px;
    padding: 20px;
    box-shadow: 0 3px 10px rgba(0,0,0,0.1);
    transition: all 0.4s;
}
```

```
.карточка:hover {
```

```
    transform: translateY(-10px);
    box-shadow: 0 15px 30px rgba(0,0,0,0.2);
}
```

---

## ЧАСТЬ 5: Чего избегать?

### ✗ Плохие практики:

1. **Слишком быстрая анимация** — глаза не успевают
2. **Слишком медленная анимация** — пользователь теряет терпение
3. **Много разных эффектов на одной странице** — отвлекает
4. **Эффекты без transition** — резкие, неприятные переходы
5. **Эффекты, которые скрывают контент** — пользователь не видит информацию

### ✓ Хорошие практики:

1. **Длительность 0.2-0.5 секунды** — оптимально для восприятия
  2. **Используйте easing функции** — ease, ease-in-out
  3. **Сохраняйте контраст** — текст должен оставаться читаемым
  4. **Тестируйте на мобильных** — touch события работают иначе
  5. **Добавляйте :focus для доступности** — для пользователей клавиатуры
- 

## ИТОГ:

Псевдокласс `:hover` — это ваш волшебный инструмент для создания интерактивных, живых веб-страниц. С его помощью вы можете:

1. **Показывать реакцию** на действия пользователя
2. **Выделять важные элементы**
3. **Создавать интуитивные интерфейсы**
4. **Добавлять элемент игры и удовольствия**

**Золотое правило:** Используйте `:hover` чтобы улучшить пользовательский опыт, а не чтобы похвастаться своими навыками CSS. Каждый эффект должен иметь смысл и помогать пользователю.

## **Запомните формулу успеха:**

text

:hover + transition = 😊 довольный пользователь

Теперь вы — настоящий волшебник CSS! Вы можете оживлять элементы, создавать интерактивные интерфейсы и делать свои сайты по-настоящему привлекательными. В следующих главах мы научимся создавать адаптивные сайты, которые красиво выглядят на любом устройстве!

- **Практикум №6: Делаем так, чтобы кнопки меню меняли цвет при наведении, а ссылки – подчёркивались.**

Дорогой друг! Сегодня мы совершим настояще волшебство — заставим меню на вашем сайте «оживать» при наведении мышки. Это как украсить ёлку гирляндами: без них красиво, а с ними — волшебно!

---

## ЧАСТЬ 1: Что мы будем делать?

Мы создадим **три разных меню**, и каждое будет по-своему реагировать на наведение курсора:

1. **Меню для садовода** — зелёные кнопки, которые становятся светлее
2. **Меню для рукодельницы** — нежные кнопки с плавными переходами
3. **Меню для кулинара** — «вкусные» кнопки, которые меняют цвет как спелые фрукты

И для каждого меню мы научимся:

- **Менять цвет фона** кнопок при наведении
  - **Добавлять подчёркивание** у ссылок
  - **Создавать плавные переходы** между состояниями
  - **Добавлять дополнительные эффекты** (тени, трансформации)
- 

## ЧАСТЬ 2: Интерактивная лаборатория

Создайте файл **практикум6-меню.html** и скопируйте этот код:

```
html
<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
<title>Практикум №6: Оживляем меню с :hover</title>
<style>

    /* ОСНОВНЫЕ СТИЛИ СТРАНИЦЫ */
    * {
        box-sizing: border-box;
        margin: 0;
        padding: 0;
    }

    body {
        font-family: 'Georgia', serif;
        line-height: 1.6;
        color: #333;
        background: linear-gradient(135deg, #f5f7fa 0%, #c3cfe2 100%);
        padding: 20px;
        min-height: 100vh;
    }

    .контейнер {
        max-width: 1200px;
        margin: 0 auto;
        background: white;
        border-radius: 20px;
        box-shadow: 0 20px 60px rgba(0,0,0,0.1);
        overflow: hidden;
    }

    .заголовок {
        background: linear-gradient(135deg, #8b4513, #a0522d);
        color: white;
        padding: 40px;
        text-align: center;
    }

    .заголовок h1 {
        margin: 0;
        font-size: 2.8rem;
        text-shadow: 2px 2px 4px rgba(0,0,0,0.3);
    }

    .заголовок p {
        font-size: 1.2rem;
        opacity: 0.9;
        max-width: 800px;
    }

```

```
margin: 20px auto 0;  
}  
  
/* БЛОК С МЕНЮ */  
.меню-блок {  
padding: 40px;  
border-bottom: 2px dashed #e0e0e0;  
}  
  
.меню-блок:last-child {  
border-bottom: none;  
}  
  
.номер-меню {  
display: inline-block;  
background: #3498db;  
color: white;  
width: 40px;  
height: 40px;  
border-radius: 50%;  
text-align: center;  
line-height: 40px;  
font-weight: bold;  
margin-bottom: 20px;  
font-size: 1.3rem;  
}  
  
h2 {  
color: #2c3e50;  
margin: 10px 0 30px 0;  
padding-bottom: 15px;  
border-bottom: 3px solid;  
}  
  
/* КОД-БЛОКИ */  
.код-блок {  
background: #2c3e50;  
color: #ecf0f1;  
padding: 25px;  
border-radius: 10px;  
font-family: 'Courier New', monospace;  
margin: 25px 0;  
overflow-x: auto;  
line-height: 1.8;
```

```
}

.css-свойство {
    color: #3498db;
}

.css-значение {
    color: #2ecc71;
}

.css-селектор {
    color: #e74c3c;
}

.css-комментарий {
    color: #95a5a6;
}

/* ИНФО-БЛОКИ */

.инфо-блок {
    background: #e8f4fc;
    padding: 20px;
    border-radius: 10px;
    margin: 20px 0;
    border-left: 5px solid #3498db;
}

.задание-блок {
    background: #fff8e1;
    padding: 20px;
    border-radius: 10px;
    margin: 20px 0;
    border-left: 5px solid #f39c12;
}

/* ПАНЕЛЬ УПРАВЛЕНИЯ */

.панель-управления {
    background: #34495e;
    color: white;
    padding: 30px;
    text-align: center;
}

.кнопка-управления {
```

```
background: #3498db;
color: white;
border: none;
padding: 15px 30px;
margin: 0 10px;
border-radius: 8px;
cursor: pointer;
font-size: 1.1rem;
transition: all 0.3s;
font-weight: bold;
}

.кнопка-управления:hover {
background: #2980b9;
transform: translateY(-3px);
box-shadow: 0 5px 15px rgba(41, 128, 185, 0.4);
}

/* МЕНЮ 1: ДЛЯ САДОВОДА */

.меню-садовода {
list-style: none;
padding: 0;
display: flex;
background: #2e8b57;
border-radius: 15px;
overflow: hidden;
box-shadow: 0 10px 25px rgba(46, 139, 87, 0.3);
}

.меню-садовода li {
flex: 1;
text-align: center;
border-right: 1px solid rgba(255,255,255,0.2);
}

.меню-садовода li:last-child {
border-right: none;
}

.меню-садовода a {
color: white;
text-decoration: none;
padding: 25px 15px;
display: block;
```

```
font-weight: bold;
font-size: 1.1rem;
transition: all 0.4s ease;
position: relative;
}

/* ЭФФЕКТ ПРИ НАВЕДЕНИИ ДЛЯ МЕНЮ 1 */

.меню-садовода a:hover {
background-color: #3cb371; /* Светлее при наведении */
}

/* ПОДЧЁРКИВАНИЕ ДЛЯ МЕНЮ 1 */

.меню-садовода a::after {
content: "";
position: absolute;
bottom: 0;
left: 50%;
width: 0;
height: 3px;
background: #ffd700; /* Золотая линия */
transition: all 0.3s;
transform: translateX(-50%);
}

.меню-садовода a:hover::after {
width: 80%; /* Линия появляется на 80% ширины */
}

/* МЕНЮ 2: ДЛЯ РУКОДЕЛЬНИЦЫ */

.меню-рукодельницы {
list-style: none;
padding: 0;
display: flex;
background: #ffebee; /* Нежно-розовый фон */
border-radius: 20px;
padding: 15px;
gap: 10px;
box-shadow: 0 10px 25px rgba(255, 182, 193, 0.3);
}

.меню-рукодельницы li {
flex: 1;
}
```

```
.меню-рукодельницы a {  
    color: #8b4513; /* Коричневый текст */  
    text-decoration: none;  
    padding: 20px 15px;  
    display: block;  
    text-align: center;  
    font-weight: bold;  
    border-radius: 50px; /* Овальные кнопки */  
    transition: all 0.5s cubic-bezier(0.175, 0.885, 0.32, 1.275);  
    background: white;  
    border: 2px solid #ffcdd2;  
    position: relative;  
    overflow: hidden;  
}  
  
/* ЭФФЕКТ ПРИ НАВЕДЕНИИ ДЛЯ МЕНЮ 2 */  
.меню-рукодельницы a:hover {  
    background: #ffcdd2; /* Розовый фон при наведении */  
    transform: translateY(-5px) scale(1.05);  
    box-shadow: 0 10px 20px rgba(255, 182, 193, 0.4);  
    color: #5d4037;  
}  
  
/* ПОДЧЁРКИВАНИЕ ДЛЯ МЕНЮ 2 (более креативное) */  
.меню-рукодельницы a::before {  
    content: " ";  
    position: absolute;  
    bottom: 0;  
    left: 0;  
    width: 100%;  
    height: 0;  
    background: #ff8a80; /* Ярко-розовая линия */  
    transition: height 0.3s;  
    z-index: -1;  
}  
  
.меню-рукодельницы a:hover::before {  
    height: 100%; /* Заполняет всю кнопку */  
}  
  
.меню-рукодельницы a span {  
    position: relative;  
    z-index: 1;  
}
```

```
/* МЕНЮ 3: ДЛЯ КУЛИНАРА */  
.меню-кулинара {  
    list-style: none;  
    padding: 0;  
    display: flex;  
    background: #fff3e0; /* Светло-оранжевый фон */  
    border-radius: 12px;  
    overflow: hidden;  
    box-shadow: 0 10px 25px rgba(255, 152, 0, 0.2);  
}  
  
.меню-кулинара li {  
    flex: 1;  
    text-align: center;  
    position: relative;  
}  
  
/* Декоративные разделители между пунктами */  
.меню-кулинара li:not(:last-child)::after {  
    content: ' \| '|;  
    position: absolute;  
    right: -10px;  
    top: 50%;  
    transform: translateY(-50%);  
    color: #ff9800;  
    font-size: 1.5rem;  
    z-index: 2;  
}  
  
.меню-кулинара a {  
    color: #e65100; /* Тёмно-оранжевый текст */  
    text-decoration: none;  
    padding: 25px 15px;  
    display: block;  
    font-weight: bold;  
    font-size: 1.1rem;  
    transition: all 0.4s;  
    position: relative;  
}  
  
/* ЭФФЕКТ ПРИ НАВЕДЕНИИ ДЛЯ МЕНЮ 3 */  
.меню-кулинара a:hover {  
    background: #ffcc80; /* Светло-оранжевый фон при наведении */
```

```
color: #bf360c;
}

/* ПОДЧЁРКИВАНИЕ ДЛЯ МЕНЮ 3 (стиль "поджаривания") */

.меню-кулинара a::after {
    content: "";
    position: absolute;
    bottom: 0;
    left: 10%;
    right: 10%;
    height: 4px;
    background: linear-gradient(90deg, transparent, #ff9800, transparent);
    border-radius: 2px;
    transform: scaleX(0);
    transition: transform 0.4s;
}

.меню-кулинара a:hover::after {
    transform: scaleX(1); /* Линия "вырастает" */
}

/* ЭФФЕКТ "ПОДЖАРИВАНИЯ" (дополнительный) */

.меню-кулинара a:hover {
    text-shadow: 0 0 10px rgba(255, 152, 0, 0.5);
}

/* ЛАБОРАТОРИЯ ЭКСПЕРИМЕНТОВ */

.лаборатория {
    padding: 40px;
    background: #f8f9fa;
    border-top: 3px solid #3498db;
}

.эксперимент-блок {
    background: white;
    padding: 30px;
    border-radius: 15px;
    margin: 30px 0;
    box-shadow: 0 10px 30px rgba(0,0,0,0.08);
}

.ползунок-группа {
    margin: 20px 0;
}
```

```
.ползунок-группа label {  
    display: block;  
    margin-bottom: 10px;  
    color: #2c3e50;  
    font-weight: bold;  
}  
  
}
```

```
input[type="range"] {  
    width: 100%;  
    margin: 10px 0;  
}  
  
}
```

```
.значение-ползунка {  
    display: inline-block;  
    min-width: 40px;  
    text-align: right;  
    font-weight: bold;  
    color: #3498db;  
}  
  
}
```

```
.цвет-выбор {  
    display: flex;  
    gap: 15px;  
    margin-top: 15px;  
    flex-wrap: wrap;  
}  
  
}
```

```
.цвет-кнопка {  
    padding: 10px 20px;  
    border: none;  
    border-radius: 5px;  
    cursor: pointer;  
    color: white;  
    font-weight: bold;  
    transition: transform 0.2s;  
}  
  
}
```

```
.цвет-кнопка:hover {  
    transform: scale(1.05);  
}  
  
}
```

```
/* ИТОГОВАЯ ЧАСТЬ */  
.итог {
```

```
padding: 40px;
background: linear-gradient(135deg, #2c3e50, #34495e);
color: white;
text-align: center;
}

.чекелист {
display: grid;
grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));
gap: 20px;
margin-top: 30px;
}

.чекелист-пункт {
background: rgba(255, 255, 255, 0.1);
padding: 20px;
border-radius: 10px;
text-align: left;
backdrop-filter: blur(10px);
border: 1px solid rgba(255, 255, 255, 0.2);
}

.чекелист-пункт.выполнено {
border-left: 5px solid #2ecc71;
}

.чекелист-пункт h4 {
color: #2ecc71;
margin-top: 0;
display: flex;
align-items: center;
gap: 10px;
}

.чекелист-пункт h4::before {
content: '✓';
display: inline-block;
width: 24px;
height: 24px;
background: #2ecc71;
border-radius: 50%;
text-align: center;
line-height: 24px;
}
```

```
</style>
</head>
<body>
  <div class="контейнер">
    <!-- ЗАГОЛОВОК -->
    <div class="заголовок">
      <h1>Практикум №6: Оживляем меню!</h1>
      <p>Наводите курсор на пункты меню ниже и наблюдайте волшебство CSS :hover</p>
    </div>

    <!-- МЕНЮ 1: ДЛЯ САДОВОДА -->
    <div class="меню-блок">
      <div class="номер-меню">1</div>
      <h2 style="border-bottom-color: #2e8b57;">Меню для садовода</h2>

      <div class="код-блок">
<span class="css-комментарий"/* Основные стили меню */</span><br>
<span class="css-селектор".меню-садовода a</span> {<br>
  &nbsp;&nbsp;<span class="css-свойство">transition</span>: <span class="css-значение"all 0.4s
  ease</span>;<br>
  &nbsp;&nbsp;<span class="css-свойство">position</span>: <span class="css-
  значение"relative</span>;<br>
}<br><br>
<span class="css-комментарий"/* 1. Меняем цвет фона при наведении */</span><br>
<span class="css-селектор".меню-садовода a:hover</span> {<br>
  &nbsp;&nbsp;<span class="css-свойство">background-color</span>: <span class="css-
  значение">#3cb371</span>;<br>
}<br><br>
<span class="css-комментарий"/* 2. Добавляем подчёркивание */</span><br>
<span class="css-селектор".меню-садовода a::after</span> {<br>
  &nbsp;&nbsp;<span class="css-свойство">content</span>: <span class="css-значение">"</span>;<br>
  &nbsp;&nbsp;<span class="css-свойство">position</span>: <span class="css-
  значение"absolute</span>;<br>
  &nbsp;&nbsp;<span class="css-свойство">bottom</span>: <span class="css-значение"0</span>;<br>
  &nbsp;&nbsp;<span class="css-свойство">width</span>: <span class="css-значение"0</span>;<br>
  &nbsp;&nbsp;<span class="css-свойство">height</span>: <span class="css-значение">3px</span>;<br>
  &nbsp;&nbsp;<span class="css-свойство">background</span>: <span class="css-
  значение">#ffd700</span>;<br>
  &nbsp;&nbsp;<span class="css-свойство">transition</span>: <span class="css-значение"all
  0.3s</span>;<br>
}<br><br>
<span class="css-селектор".меню-садовода a:hover::after</span> {<br>
  &nbsp;&nbsp;<span class="css-свойство">width</span>: <span class="css-значение">80%</span>;<br>
}
```

```

        </div>

        <div class="инфо-блок">
            <strong>📝 Как это работает:</strong>
            <p>1. При наведении курсора фон кнопки меняется с #2e8b57 на #3cb371
(становится светлее).</p>
            <p>2. Снизу появляется золотая линия, которая плавно расширяется до 80%
ширины кнопки.</p>
            <p>3. <code>transition: all 0.4s ease</code> делает переход плавным.</p>
        </div>

        <!-- ДЕМО МЕНЮ 1 -->
        <div style="margin: 40px 0; text-align: center;">
            <h3 style="color: #2e8b57; margin-bottom: 20px;">Демонстрация (наведите
курсор):</h3>
            <ul class="меню-садовода">
                <li><a href="#home"> Главная</a></li>
                <li><a href="#garden"> Мой сад</a></li>
                <li><a href="#flowers"> Цветы</a></li>
                <li><a href="#vegetables"> Овощи</a></li>
                <li><a href="#tips"> Советы</a></li>
            </ul>
        </div>
    </div>

    <!-- МЕНЮ 2: ДЛЯ РУКОДЕЛЬНИЦЫ -->
    <div class="меню-блок">
        <div class="номер-меню">2</div>
        <h2 style="border-bottom-color: #f48fb1;">Меню для рукодельницы</h2>

        <div class="код-блок">
<span class="css-комментарий"/* Более сложная анимация */</span><br>
<span class="css-селектор">.меню-рукодельницы a</span> {<br>
    &nbsp;&nbsp;<span class="css-свойство">transition</span>: <span class="css-значение">all 0.5s cubic-
    bezier(0.175, 0.885, 0.32, 1.275)</span>;<br>
    &nbsp;&nbsp;<span class="css-свойство">position</span>: <span class="css-
    значение">relative</span>;<br>
    &nbsp;&nbsp;<span class="css-свойство">overflow</span>: <span class="css-
    значение">hidden</span>;<br>
}<br><br>
<span class="css-комментарий"/* 1. Меняем цвет и добавляем движение */</span><br>
<span class="css-селектор">.меню-рукодельницы a:hover</span> {<br>
    &nbsp;&nbsp;<span class="css-свойство">background</span>: <span class="css-
    значение">#ffcdd2</span>;<br>
}

```

```

<span class="css-свойство">transform</span>: <span class="css-значение">translateY(-5px) scale(1.05)</span>;<br>
<span class="css-свойство">box-shadow</span>: <span class="css-значение">0 10px 20px rgba(255, 182, 193, 0.4)</span>;<br>
}<br><br>
<span class="css-комментарий">/* 2. Креативное "подчёркивание" - заполнение цветом
```

\* /</span><br>

```

<span class="css-селектор">.меню-рукодельницы a::before {<br>
<span class="css-свойство">content</span>: <span class="css-значение">>"</span>;<br>
<span class="css-свойство">position</span>: <span class="css-значение">absolute</span>;<br>
<span class="css-свойство">bottom</span>: <span class="css-значение">0</span>;<br>
<span class="css-свойство">width</span>: <span class="css-значение">100%</span>;<br>
<span class="css-свойство">height</span>: <span class="css-значение">0</span>;<br>
<span class="css-свойство">background</span>: <span class="css-значение">#ff8a80</span>;<br>
<span class="css-свойство">transition</span>: <span class="css-значение">height 0.3s</span>;<br>
<span class="css-свойство">z-index</span>: <span class="css-значение">-1</span>;<br>
}<br><br>
<span class="css-селектор">.меню-рукодельницы a:hover::before {<br>
<span class="css-свойство">height</span>: <span class="css-значение">100%</span>;<br>
}
```

}

```
</div>
```

```

<div class="инфо-блок">
    <strong>🎨 Особенности этого меню:</strong>
    <p>1. <code>cubic-bezier(0.175, 0.885, 0.32, 1.275)</code> — специальная
функция для "пружинной" анимации.</p>
    <p>2. Кнопка не только меняет цвет, но и "подпрыгивает" (<code>translateY(-5px)</code>) и немного увеличивается (<code>scale(1.05)</code>).</p>
    <p>3. Вместо линии снизу — плавное заполнение цветом снизу вверх.</p>
</div>
```

```

<!-- ДЕМО МЕНЮ 2 -->
<div style="margin: 40px 0; text-align: center;">
    <h3 style="color: #d81b60; margin-bottom: 20px;">Демонстрация (наведите
курсор):</h3>
    <ul class="меню-рукодельницы">
        <li><a href="#home"><span>🏡 Главная</span></a></li>
        <li><a href="#works"><span>🎁 Мои работы</span></a></li>
        <li><a href="#patterns"><span>📐 Схемы</span></a></li>
        <li><a href="#materials"><span>🛍️ Материалы</span></a></li>
    
```

```
<li><a href="#masterclass"><span>👤 Мастер-классы</span></a></li>
</ul>
</div>
</div>

<!-- МЕНЮ 3: ДЛЯ КУЛИНАРА -->
<div class="меню-блок">
    <div class="номер-меню">3</div>
    <h2 style="border-bottom-color: #ff9800;">Меню для кулинара</h2>

    <div class="код-блок">
        <span class="css-комментарий">/* Эффект "поджаривания" */</span><br>
        <span class="css-селектор">.меню-кулинара a</span> {<br>
            &nbsp;&nbsp;<span class="css-свойство">transition</span>: <span class="css-значение">all
            0.4s</span>;<br>
            &nbsp;&nbsp;<span class="css-свойство">position</span>: <span class="css-значение">relative</span>;<br>
        }<br><br>
        <span class="css-комментарий">/* 1. Меняем цвет фона и текста */</span><br>
        <span class="css-селектор">.меню-кулинара a:hover</span> {<br>
            &nbsp;&nbsp;<span class="css-свойство">background</span>: <span class="css-значение">#ffcc80</span>;<br>
            &nbsp;&nbsp;<span class="css-свойство">color</span>: <span class="css-значение">#bf360c</span>;<br>
            &nbsp;&nbsp;<span class="css-свойство">text-shadow</span>: <span class="css-значение">0 0 10px
            rgba(255, 152, 0, 0.5)</span>;<br>
        }<br><br>
        <span class="css-комментарий">/* 2. Линия, которая "вырастает" */</span><br>
        <span class="css-селектор">.меню-кулинара a::after</span> {<br>
            &nbsp;&nbsp;<span class="css-свойство">content</span>: <span class="css-значение">"</span>;<br>
            &nbsp;&nbsp;<span class="css-свойство">position</span>: <span class="css-значение">absolute</span>;<br>
            &nbsp;&nbsp;<span class="css-свойство">bottom</span>: <span class="css-значение">0</span>;<br>
            &nbsp;&nbsp;<span class="css-свойство">left</span>: <span class="css-значение">10%</span>;<br>
            &nbsp;&nbsp;<span class="css-свойство">right</span>: <span class="css-значение">10%</span>;<br>
            &nbsp;&nbsp;<span class="css-свойство">height</span>: <span class="css-значение">4px</span>;<br>
            &nbsp;&nbsp;<span class="css-свойство">background</span>: <span class="css-значение">linear-
            gradient(90deg, transparent, #ff9800, transparent)</span>;<br>
            &nbsp;&nbsp;<span class="css-свойство">transform</span>: <span class="css-значение">transform
            scaleX(0)</span>;<br>
            &nbsp;&nbsp;<span class="css-свойство">transition</span>: <span class="css-значение">transform
            0.4s</span>;<br>
        }<br><br>
        <span class="css-селектор">.меню-кулинара a:hover::after</span> {<br>
```

```

<span class="css-свойство">transform</span>: <span class="css-значение">scaleX(1)</span>;<br>
}

</div>

<div class="инфо-блок">
    <strong>💡 Кулинарные фишки:</strong>
    <p>1. Градиентная линия: <code>linear-gradient(90deg, transparent, #ff9800, transparent)</code> создаёт эффект "светящейся" линии.</p>
    <p>2. <code>transform: scaleX(0)</code> скрывает линию, а <code>scaleX(1)</code> показывает её с эффектом роста.</p>
    <p>3. <code>text-shadow</code> добавляет лёгкое свечение тексту, как будто он "подрумянивается".</p>
</div>

<!-- ДЕМО МЕНЮ 3 -->
<div style="margin: 40px 0; text-align: center;">
    <h3 style="color: #ef6c00; margin-bottom: 20px;">Демонстрация (наведите курсор):</h3>
    <ul class="меню-кулинара">
        <li><a href="#home">🌐 Главная</a></li>
        <li><a href="#recipes">📜 Рецепты</a></li>
        <li><a href="#video">🎥 Видеоуроки</a></li>
        <li><a href="#secrets">🧠 Секреты</a></li>
        <li><a href="#contacts">📞 Контакты</a></li>
    </ul>
</div>
</div>

<!-- ЛАБОРАТОРИЯ: СОЗДАЙ СВОЁ МЕНЮ -->
<div class="лаборатория">
    <h2 style="text-align: center; color: #2c3e50;">🔬 Лаборатория: Создай своё меню!
</h2>

    <div class="задание-блок">
        <h3 style="color: #f39c12;">Задание: Создайте меню для вашего хобби</h3>
        <p>Используйте панель ниже, чтобы настроить цвета и эффекты. Затем скопируйте готовый CSS-код!</p>
    </div>

    <!-- ПАНЕЛЬ НАСТРОЕК -->
    <div class="эксперимент-блок">
        <h3>Настройки вашего меню:</h3>

```

```
<div class="ползунок-группа">
    <label>Скорость анимации: <span class="значение-ползунка"
id="скоростьЗначение">0.4</span>s</label>
    <input type="range" id="скоростьПолзунок" min="0.1" max="2" step="0.1"
value="0.4" oninput="обновитьМеню()">
</div>

<div class="ползунок-группа">
    <label>Высота подчёркивания: <span class="значение-ползунка"
id="высотаЗначение">3</span>px</label>
    <input type="range" id="высотаПолзунок" min="1" max="10" step="1"
value="3" oninput="обновитьМеню()">
</div>

<div>
    <label style="display: block; margin-bottom: 10px; font-weight: bold;">Цвет
фона при наведении:</label>
    <div class="цвет-выбор">
        <button class="цвет-кнопка" style="background: #3cb371;">Зелёный</button>
        <button class="цвет-кнопка" style="background: #3498db;">Синий</button>
        <button class="цвет-кнопка" style="background: #e74c3c;">Красный</button>
        <button class="цвет-кнопка" style="background: #f39c12;">Оранжевый</button>
        <button class="цвет-кнопка" style="background: #9b59b6;">Фиолетовый</button>
    </div>
</div>

<div style="margin-top: 20px;">
    <label style="display: block; margin-bottom: 10px; font-weight: bold;">Цвет
подчёркивания:</label>
    <div class="цвет-выбор">
        <button class="цвет-кнопка" style="background: #ffd700;">Золотой</button>
        <button class="цвет-кнопка" style="background: #2ecc71;">Светло-зелёный</button>
        <button class="цвет-кнопка" style="background: #e74c3c;">Красный</button>
        <button class="цвет-кнопка" style="background: #3498db;">Синий</button>
```

```
<button class="цвет-кнопка" style="background: #ffffff; color: #333;"  
onclick="выбратьЦвет('underlineColor', '#ffffff')">Белый</button>  
</div>  
</div>
```

Скопировать CSS-код</button>  
<button class="кнопка-управления" onclick="скопироватьКод()">

Сбросить настройки</button>  
</div>  
</div>

```
<!-- ПРЕДПРОСМОТР -->  
<div class="эксперимент-блок">  
    <h3>Предпросмотр вашего меню:</h3>  
    <div id="предпросмотрМеню" style="margin: 30px 0;">  
        <!-- Сюда вставится сгенерированное меню -->  
    </div>  
</div>
```

```
<!-- СГЕНЕРИРОВАННЫЙ КОД -->  
<div class="эксперимент-блок">  
    <h3>Ваш CSS-код:</h3>  
    <div class="код-блок" id="сгенерированныйКод">
```

/\* Ваш CSS-код появится здесь после настройки \*/  
</div>  
</div>  
</div>

```
<!-- ПАНЕЛЬ УПРАВЛЕНИЯ -->  
<div class="панель-управления">  
    <h3 style="color: white; margin-top: 0;">Быстрые эффекты</h3>  
    <p style="color: #ecf0f1; margin-bottom: 25px;">Нажмите на кнопку, чтобы применить  
готовый эффект ко всем меню</p>
```

<button class="кнопка-управления" onclick="применитьЭффект('pulse')">

Пульсация</button>  
<button class="кнопка-управления" onclick="применитьЭффект('bounce')">

Подпрыгивание</button>  
<button class="кнопка-управления" onclick="применитьЭффект('glow')">

Свечение</button>  
<button class="кнопка-управления" onclick="применитьЭффект('shake')">

Дрожание</button>

```
<button class="кнопка-управления" onclick="сброситьЭффекты()" style="background:#e74c3c;">❌ Сбросить эффекты</button>
</div>

<!-- ИТОГ -->
<div class="итог">
    <h2 style="color: white;">🎉 Поздравляю! Вы научились оживлять меню!</h2>

    <div class="чекалист">
        <div class="чекалист-пункт выполнено">
            <h4>Менять цвет кнопок</h4>
            <p>Теперь вы умеете менять background-color при наведении</p>
        </div>

        <div class="чекалист-пункт выполнено">
            <h4>Добавлять подчёркивание</h4>
            <p>Вы научились создавать красивые линии с помощью ::after</p>
        </div>

        <div class="чекалист-пункт выполнено">
            <h4>Создавать плавные переходы</h4>
            <p>transition — ваш лучший друг для плавных анимаций</p>
        </div>

        <div class="чекалист-пункт выполнено">
            <h4>Экспериментировать с эффектами</h4>
            <p>Вы можете комбинировать transform, box-shadow и другие свойства</p>
        </div>
    </div>
</div>

<div style="margin-top: 40px;">
    <h3 style="color: #2ecc71;">Ваше домашнее задание:</h3>
    <p style="font-size: 1.2rem; max-width: 800px; margin: 20px auto;">
        Создайте меню для вашего собственного сайта! Используйте цвета, которые соответствуют тематике вашего хобби.
    </p>
    <button class="кнопка-управления" onclick="alert('Удачи в создании вашего уникального меню! Помните: практика — ключ к мастерству. 🔑')" style="background: #2ecc71; margin-top: 20px;">
        Принять задание!
    </button>
</div>
</div>
```

```
</div>

<script>
    // ПЕРЕМЕННЫЕ ДЛЯ НАСТРОЕК
    let текущиеНастройки = {
        скорость: 0.4,
        высотаПодчеркивания: 3,
        цветПриНаведении: '#3cb371',
        цветПодчеркивания: '#ffd700'
    };

    // ФУНКЦИЯ ОБНОВЛЕНИЯ ПРЕДПРОСМОТРА
    function обновитьМеню() {
        // Получаем значения с ползунков
        текущиеНастройки.скорость =
parseFloat(document.getElementById('скоростьПолзунок').value);
        текущиеНастройки.высотаПодчеркивания =
parseInt(document.getElementById('высотаПолзунок').value);

        // Обновляем значения на экране
        document.getElementById('скоростьЗначение').textContent =
текущиеНастройки.скорость;
        document.getElementById('высотаЗначение').textContent =
текущиеНастройки.высотаПодчеркивания;

        // Создаём CSS для предпросмотра
        const css = `
.mеню-предпросмотр {
            list-style: none;
            padding: 0;
            display: flex;
            background: #2c3e50;
            border-radius: 15px;
            overflow: hidden;
            margin: 0;
        }

.mеню-предпросмотр li {
            flex: 1;
            text-align: center;
        }

.mеню-предпросмотр a {
            color: white;

```

```

        text-decoration: none;
        padding: 20px 15px;
        display: block;
        font-weight: bold;
        transition: all ${текущиеНастройки.скорость}s ease;
        position: relative;
    }

.меню-предпросмотр a:hover {
    background-color: ${текущиеНастройки.цветПриНаведении};
}

.меню-предпросмотр a::after {
    content: "";
    position: absolute;
    bottom: 0;
    left: 50%;
    width: 0;
    height: ${текущиеНастройки.высотаПодчеркивания}px;
    background: ${текущиеНастройки.цветПодчеркивания};
    transition: all ${текущиеНастройки.скорость / 2}s;
    transform: translateX(-50%);
}

.меню-предпросмотр a:hover::after {
    width: 80%;
}

`;

// Создаём HTML для меню
const html = `
<ul class="меню-предпросмотр">
    <li><a href="#">Главная</a></li>
    <li><a href="#">О проекте</a></li>
    <li><a href="#">Услуги</a></li>
    <li><a href="#">Портфолио</a></li>
    <li><a href="#">Контакты</a></li>
</ul>
`;

// Вставляем стили и HTML
const предпросмотр = document.getElementById('предпросмотрМеню');
предпросмотр.innerHTML = `<style>${css}</style>${html}`;

```

```
// Обновляем код для копирования  
обновитьСгенерированныйКод();  
}  
  
// ФУНКЦИЯ ВЫБОРА ЦВЕТА  
function выбратьЦвет(тип, цвет) {  
    if (тип === 'hoverColor') {  
        текущиеНастройки.цветПриНаведении = цвет;  
    } else if (тип === 'underlineColor') {  
        текущиеНастройки.цветПодчеркивания = цвет;  
    }  
    обновитьМеню();  
}  
  
// ФУНКЦИЯ ОБНОВЛЕНИЯ СГЕНЕРИРОВАННОГО КОДА  
function обновитьСгенерированныйКод() {  
    const код = `/* CSS для вашего меню */  
  
/* Основные стили меню */  
.ваше-меню {  
    list-style: none;  
    padding: 0;  
    display: flex;  
    background: #2c3e50;  
    border-radius: 15px;  
    overflow: hidden;  
}  
  
.ваше-меню li {  
    flex: 1;  
    text-align: center;  
}  
  
.ваше-меню a {  
    color: white;  
    text-decoration: none;  
    padding: 20px 15px;  
    display: block;  
    font-weight: bold;  
    transition: all ${текущиеНастройки.скорость}s ease;  
    position: relative;  
}  
  
/* 1. Меняем цвет фона при наведении */
```

```

.ваше-меню a:hover {
    background-color: ${текущиеНастройки.цветПриНаведении};
}

/* 2. Добавляем подчёркивание */
.ваše-меню a::after {
    content: "";
    position: absolute;
    bottom: 0;
    left: 50%;
    width: 0;
    height: ${текущиеНастройки.высотаПодчеркивания}px;
    background: ${текущиеНастройки.цветПодчеркивания};
    transition: all ${текущиеНастройки.скорость / 2}s;
    transform: translateX(-50%);
}

.ваše-меню a:hover::after {
    width: 80%;
};

};

document.getElementById('сгенерированныйКод').textContent = код;
}

// ФУНКЦИЯ КОПИРОВАНИЯ КОДА
function скопироватьКод() {
    const код = document.getElementById('сгенерированныйКод').textContent;
    navigator.clipboard.writeText(код)
        .then(() => alert('✓ CSS-код скопирован в буфер обмена!\n\nТеперь вы можете
вставить его в свой файл style.css'))
        .catch(err => alert('✗ Не удалось скопировать код: ' + err));
}

// ФУНКЦИЯ СБРОСА НАСТРОЕК
function сброситьНастройки() {
    document.getElementById('скоростьПолзунок').value = 0.4;
    document.getElementById('высотаПолзунок').value = 3;
    текущиеНастройки = {
        скорость: 0.4,
        высотаПодчеркивания: 3,
        цветПриНаведении: '#3cb371',
        цветПодчеркивания: '#ffd700'
    };
    обновитьМеню();
}

```

```
}

// ФУНКЦИИ ДЛЯ БЫСТРЫХ ЭФФЕКТОВ
function применитьЭффект(тип) {
    let css = ``;

    switch(тип) {
        case 'pulse':
            css = `
                .меню-садовода a:hover,
                .меню-рукодельницы a:hover,
                .меню-кулинара a:hover {
                    animation: pulse 0.5s infinite alternate;
                }

                @keyframes pulse {
                    from { transform: scale(1); }
                    to { transform: scale(1.05); }
                };
                break;

        case 'bounce':
            css = `
                .меню-садовода a:hover,
                .меню-рукодельницы a:hover,
                .меню-кулинара a:hover {
                    animation: bounce 0.5s;
                }

                @keyframes bounce {
                    0%, 100% { transform: translateY(0); }
                    50% { transform: translateY(-10px); }
                };
                break;

        case 'glow':
            css = `
                .меню-садовода a:hover,
                .меню-рукодельницы a:hover,
                .меню-кулинара a:hover {
                    box-shadow: 0 0 15px ${текущиеНастройки.цветПриНаведении};
                };
                break;
    }
}
```

```

        case 'shake':
            css = `
                .меню-садовода a:hover,
                .меню-рукодельницы a:hover,
                .меню-кулинара a:hover {
                    animation: shake 0.5s;
                }

                @keyframes shake {
                    0%, 100% { transform: translateX(0); }
                    25% { transform: translateX(-5px); }
                    75% { transform: translateX(5px); }
                };
            break;
    }

    // Добавляем стили на страницу
    const style = document.createElement('style');
    style.id = 'эффекты';
    style.textContent = css;

    // Удаляем старые эффекты
    const старыеЭффекты = document.getElementById('эффекты');
    if (старыеЭффекты) старыеЭффекты.remove();

    document.head.appendChild(style);

    alert(`Эффект "${тип}" применён ко всем меню!`);

}

function сброситьЭффекты() {
    const эффекты = document.getElementById('эффекты');
    if (эффекты) эффекты.remove();
    alert('Эффекты сброшены!');
}

// ИНИЦИАЛИЗАЦИЯ ПРИ ЗАГРУЗКЕ
document.addEventListener('DOMContentLoaded', function() {
    // Инициализируем предпросмотр
    обновитьМеню();

    // Делаем все демо-ссылки неактивными
    document.querySelectorAll('.меню-садовода a, .меню-рукодельницы a, .меню-кулинара a').forEach(link => {

```

```
link.addEventListener('click', function(e) {
    e.preventDefault();
    alert('В демо-версии ссылки не работают. В реальном сайте они бы вели
на соответствующие страницы!');
});
});
});
};

</script>
</body>
</html>
```

---

## ЧАСТЬ 3: Ключевые техники, которые вы освоили

### 1. Меняем цвет фона кнопок:

css

```
/* Исходный цвет */
.menu a {
    background-color: #2e8b57;
}

/* Цвет при наведении */
.menu a:hover {
    background-color: #3cb371; /* Светлее! */
}
```

### 2. Добавляем подчёркивание с помощью ::after:

css

```
.menu a {
    position: relative; /* Важно! */
}

.menu a::after {
    content: ""; /* Пустой элемент */
    position: absolute;
    bottom: 0;
    left: 50%;
    width: 0; /* Сначала не видна */
```

```
height: 3px;  
background: #ffd700;  
transition: width 0.3s; /* Плавное появление */  
transform: translateX(-50%);  
}  
  
.menu a:hover::after {
```

```
width: 80%; /* Появляется при наведении */  
}
```

### 3. Плавные переходы с transition:

css

```
.menu a {  
    transition: all 0.4s ease;  
    /* all - все свойства */  
    /* 0.4s - длительность */  
    /* ease - тип анимации */  
}
```

### 4. Комбинированные эффекты:

css

```
.menu a:hover {  
    background-color: #3cb371; /* Меняем цвет */  
    transform: translateY(-5px); /* Поднимаем */  
    box-shadow: 0 5px 15px rgba(0,0,0,0.2); /* Добавляем тень */  
}
```

---

## ЧАСТЬ 4: Готовые рецепты для ваших проектов

### Рецепт 1: Простое и элегантное меню

css

```
/* Для сайта о литературе или искусстве */  
.literature-menu a:hover {  
    background-color: #8b4513; /* Коричневый */  
    color: #fff8e1; /* Слоновая кость */
```

```
}
```

```
.literature-menu a::after {  
    background: #d2b48c; /* Бежевая линия */  
}
```

## Рецепт 2: Яркое и современное меню

css

```
/* Для блога или портфолио */  
.modern-menu a:hover {  
    background: linear-gradient(135deg, #3498db, #2ecc71); /* Градиент! */  
    transform: scale(1.05);  
}  
  
.modern-menu a::after {  
    background: white; /* Белая линия */  
    height: 2px;  
}
```

## Рецепт 3: Нежное меню для личного блога

css

```
/* Для дневника или личного сайта */  
.soft-menu a {  
    background: rgba(255,255,255,0.1); /* Полупрозрачный */  
    border-radius: 20px;  
}  
  
.soft-menu a:hover {  
    background: rgba(255,255,255,0.3);  
    backdrop-filter: blur(10px); /* Эффект размытия */  
}  
  
.soft-menu a::after {  
    display: none; /* Без линии, только изменение фона */  
}
```

---

## ЧАСТЬ 5: Частые ошибки и как их избежать

### ✖ Ошибка 1: Эффект слишком резкий

css

```
/* ПЛОХО: */  
.menu a:hover {  
    background-color: red; /* Резкая смена */  
}
```

```
/* ХОРОШО: */  
.menu a {  
    transition: background-color 0.3s;  
}  
.menu a:hover {  
    background-color: red; /* Теперь плавно */  
}
```

### ✖ Ошибка 2: Подчёркивание не по центру

css

```
/* ПЛОХО: */  
.menu a::after {  
    left: 0; /* Прижато к левому краю */  
}
```

```
/* ХОРОШО: */  
.menu a::after {  
    left: 50%;  
    transform: translateX(-50%); /* По центру! */  
}
```

### ✖ Ошибка 3: Эффекты для мобильных устройств

css

```
/* ПЛОХО: Только :hover */  
.menu a:hover { ... }  
  
/* ХОРОШО: Добавляем :active для мобильных */  
.menu a:hover,  
.menu a:active { ... }
```

```
.menu a:active { /* Однаковые стили */
    background-color: #3cb371;
}
```

---

## ЧАСТЬ 6: Практическое задание

### Задание «Создай меню для своего хобби»

#### 1. Выберите тему:

- Садоводство (зелёные тона)
- Вязание (розовые, бежевые тона)
- Кулинария (оранжевые, красные тона)
- Фотография (чёрно-белые, серые тона)
- Путешествия (синие, бирюзовые тона)

#### 2. Создайте HTML-строктуру:

html

```
<ul class="мое-меню">
    <li><a href="#">Главная</a></li>
    <li><a href="#">Обо мне</a></li>
    <li><a href="#">Моё хобби</a></li>
    <li><a href="#">Галерея</a></li>
    <li><a href="#">Контакты</a></li>
</ul>
```

#### 4. Добавьте CSS-стили:

- Фон меню
- Цвета кнопок в обычном состоянии
- Цвета кнопок при наведении (:hover)
- Подчёркивание (::after)
- Плавные переходы (transition)

#### 5. Экспериментируйте!

- Попробуйте разные цвета
  - Измените скорость анимации
  - Добавьте дополнительные эффекты (тень, трансформацию)
-

## ИТОГ:

Поздравляю! Теперь вы умеете создавать живые, интерактивные меню для своих сайтов. Вы освоили:

1. **:hover** — волшебный псевдокласс для реакции на наведение
2. **::after** — создание декоративных элементов без изменения HTML
3. **transition** — плавные анимации между состояниями
4. **Комбинирование эффектов** — цвет, движение, тени, подчёркивание

**Совет:** Начните с простого меню с одним эффектом, а потом постепенно добавляйте сложности. Помните — лучшее меню то, которое удобно использовать!

### Формула успеха:

text

:hover + ::after + transition = ✨ Волшебное меню!

Теперь вы можете оживить любую навигацию на своих сайтах. В следующей главе мы научимся работать с изображениями — делать их адаптивными и красиво оформленными!

## Часть 4: Финальные штрихи и большие возможности

### ● Глава 10: Работа с ИЗОБРАЖЕНИЯМИ.

- Как сделать изображение адаптивным (`max-width: 100%`)?

Дорогой друг! Сегодня мы научимся одной из самых важных магий в веб-дизайне — заставим изображения подстраиваться под любой размер экрана! Это как волшебная одежда, которая сама подстраивается под любого человека. На большом экране компьютера — картинка большая, на маленьком экране телефона — картинка уменьшается, но остаётся красивой и чёткой!

---

### ЧАСТЬ 1: Почему это так важно?

Представьте, что вы отправляете фотографию своего сада друзьям. На компьютере они видят её во всей красе, а на телефоне... о ужас! Картинка не помещается, появляется полоса прокрутки, нужно двигать экран влево-вправо. Неудобно!

#### Проблема без адаптивных изображений:

- На мобильных телефонах картинки выходят за границы экрана
- Появляются горизонтальные полосы прокрутки
- Пользователи должны масштабировать и двигать экран
- Сайт выглядит неаккуратно и непрофессионально

**Решение:** `max-width: 100%` — это волшебное свойство, которое говорит изображению: «Ты можешь быть любого размера, но не больше, чем твой родительский контейнер!»

---

## ЧАСТЬ 2: Интерактивная лаборатория адаптивных изображений

Создайте файл **адаптивные-изображения.html** и скопируйте этот код:

```
html
<!DOCTYPE html>
<html lang="ru">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<title>Волшебство адаптивных изображений: max-width: 100%</title>
<style>
/* ОСНОВНЫЕ СТИЛИ СТРАНИЦЫ */
* {
    box-sizing: border-box;
    margin: 0;
    padding: 0;
}

body {
    font-family: 'Georgia', serif;
    line-height: 1.6;
    color: #333;
    background: linear-gradient(135deg, #f5f7fa 0%, #c3cfe2 100%);
    padding: 20px;
    min-height: 100vh;
}

.контейнер {
    max-width: 1200px;
    margin: 0 auto;
    background: white;
    border-radius: 20px;
    box-shadow: 0 20px 60px rgba(0,0,0,0.1);
    overflow: hidden;
}

.заголовок {
    background: linear-gradient(135deg, #8b4513, #a0522d);
    color: white;
    padding: 40px;
```

```
    text-align: center;  
}  
  
.заголовок h1 {  
margin: 0;  
font-size: 2.8rem;  
text-shadow: 2px 2px 4px rgba(0,0,0,0.3);  
}  
  
.заголовок p {  
font-size: 1.2rem;  
opacity: 0.9;  
max-width: 800px;  
margin: 20px auto 0;  
}  
  
/* ОСНОВНОЙ КОНТЕНТ */  
.содержание {  
padding: 40px;  
}  
  
h2 {  
color: #2c3e50;  
margin: 30px 0 20px 0;  
padding-bottom: 15px;  
border-bottom: 3px solid #3498db;  
}  
  
h3 {  
color: #2c3e50;  
margin: 25px 0 15px 0;  
}  
  
/* КОД-БЛОКИ */  
.код-блок {  
background: #2c3e50;  
color: #ecf0f1;  
padding: 25px;  
border-radius: 10px;  
font-family: 'Courier New', monospace;  
margin: 25px 0;  
overflow-x: auto;  
line-height: 1.8;  
}
```

```
.css-свойство {
    color: #3498db;
}

.css-значение {
    color: #2ecc71;
}

.css-селектор {
    color: #e74c3c;
}

.css-комментарий {
    color: #95a5a6;
}

/* ИНФО-БЛОКИ */
.инфо-блок {
    background: #e8f4fc;
    padding: 20px;
    border-radius: 10px;
    margin: 20px 0;
    border-left: 5px solid #3498db;
}

.предупреждение {
    background: #ffea7;
    padding: 20px;
    border-radius: 10px;
    margin: 20px 0;
    border-left: 5px solid #f39c12;
}

.успех {
    background: #d5f4e6;
    padding: 20px;
    border-radius: 10px;
    margin: 20px 0;
    border-left: 5px solid #2ecc71;
}

/* ЭКСПЕРИМЕНТАЛЬНАЯ ОБЛАСТЬ */
.эксперимент {
```

```
background: #f8f9fa;
padding: 30px;
border-radius: 15px;
margin: 30px 0;
border: 2px solid #e9ecf;
}
```

```
.эксперимент-заголовок {
display: flex;
align-items: center;
gap: 15px;
margin-bottom: 25px;
}
```

```
.номер-эксперимента {
display: inline-block;
background: #3498db;
color: white;
width: 40px;
height: 40px;
border-radius: 50%;
text-align: center;
line-height: 40px;
font-weight: bold;
font-size: 1.3rem;
}
```

```
/* ПАНЕЛЬ УПРАВЛЕНИЯ */
.панель-управления {
background: #34495e;
color: white;
padding: 30px;
border-radius: 10px;
margin: 30px 0;
text-align: center;
}
```

```
.кнопка-управления {
background: #3498db;
color: white;
border: none;
padding: 15px 30px;
margin: 0 10px 10px 0;
border-radius: 8px;
```

```
cursor: pointer;
font-size: 1.1rem;
transition: all 0.3s;
font-weight: bold;
}

.кнопка-управления:hover {
background: #2980b9;
transform: translateY(-3px);
box-shadow: 0 5px 15px rgba(41, 128, 185, 0.4);
}

/* СЕТКА ДЛЯ ИЗОБРАЖЕНИЙ */
.сетка-изображений {
display: grid;
grid-template-columns: repeat(auto-fit, minmax(300px, 1fr));
gap: 30px;
margin: 30px 0;
}

.изображение-блок {
background: white;
padding: 20px;
border-radius: 10px;
box-shadow: 0 5px 15px rgba(0,0,0,0.1);
border: 2px solid #e9ecf;
transition: transform 0.3s;
}

.изображение-блок:hover {
transform: translateY(-5px);
border-color: #3498db;
}

/* СТИЛИ ДЛЯ ИЗОБРАЖЕНИЙ */
.изображение-фиксированное {
width: 800px; /* Фиксированная ширина - ПЛОХО! */
border: 3px solid #e74c3c;
}

.изображение-адаптивное {
max-width: 100%; /* Волшебное свойство! */
height: auto; /* Сохраняем пропорции */
border: 3px solid #2ecc71;
```

```
}

.изображение-высокое {
    max-width: 100%;
    height: 400px; /* Фиксированная высота - может исказить */
    border: 3px solid #f39c12;
    object-fit: cover; /* Спасает ситуацию */
}

/* КОНТЕЙНЕРЫ ДЛЯ ЭКСПЕРИМЕНТОВ */

.контейнер-малый {
    width: 300px;
    border: 2px dashed #3498db;
    padding: 15px;
    margin: 20px 0;
}

.контейнер-средний {
    width: 500px;
    border: 2px dashed #9b59b6;
    padding: 15px;
    margin: 20px 0;
}

.контейнер-большой {
    width: 800px;
    border: 2px dashed #2ecc71;
    padding: 15px;
    margin: 20px 0;
}

/* ЭМУЛЯТОР РАЗНЫХ УСТРОЙСТВ */

.эмулатор-устройств {
    display: flex;
    flex-wrap: wrap;
    gap: 20px;
    margin: 30px 0;
    justify-content: center;
}

.устройство {
    background: white;
    border-radius: 20px;
    padding: 20px;
```

```
        box-shadow: 0 10px 30px rgba(0,0,0,0.1);
        text-align: center;
        border: 3px solid;
    }

.смартфон {
    width: 300px;
    border-color: #3498db;
}

.планшет {
    width: 500px;
    border-color: #9b59b6;
}

.ноутбук {
    width: 800px;
    border-color: #2ecc71;
}

.название-устройства {
    font-weight: bold;
    margin-bottom: 15px;
    color: #2c3e50;
}

/* ПРАКТИЧЕСКИЕ ПРИМЕРЫ */

.пример-галерея {
    display: grid;
    grid-template-columns: repeat(auto-fill, minmax(250px, 1fr));
    gap: 20px;
    margin: 30px 0;
}

.галерея-фото {
    border-radius: 10px;
    overflow: hidden;
    box-shadow: 0 5px 15px rgba(0,0,0,0.1);
    transition: transform 0.3s;
}

.галерея-фото:hover {
    transform: scale(1.05);
}
```

```
.галерея-фото img {
    width: 100%;
    height: 200px;
    object-fit: cover;
    display: block;
}

.подпись-галереи {
    padding: 15px;
    background: white;
    text-align: center;
    font-style: italic;
    color: #666;
}

/* ФУТЕР С ИТОГОМ */
.итог {
    background: linear-gradient(135deg, #2c3e50, #34495e);
    color: white;
    padding: 40px;
    text-align: center;
    margin-top: 40px;
}

.чекелист {
    display: grid;
    grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));
    gap: 20px;
    margin-top: 30px;
}

.чекелист-пункт {
    background: rgba(255, 255, 255, 0.1);
    padding: 20px;
    border-radius: 10px;
    text-align: left;
    backdrop-filter: blur(10px);
    border: 1px solid rgba(255, 255, 255, 0.2);
}

.чекелист-пункт h4 {
    color: #2ecc71;
    margin-top: 0;
```

```
display: flex;
align-items: center;
gap: 10px;
}

.чекелист-пункт h4::before {
  content: '✓';
  display: inline-block;
  width: 24px;
  height: 24px;
  background: #2ecc71;
  border-radius: 50%;
  text-align: center;
  line-height: 24px;
}

/* АДАПТИВНЫЕ МЕДИА-ЗАПРОСЫ ДЛЯ ДЕМОНСТРАЦИИ */

@media (max-width: 768px) {
  .заголовок h1 {
    font-size: 2rem;
  }

  .содержание {
    padding: 20px;
  }

  .контейнер-большой,
  .контейнер-средний,
  .контейнер-малый {
    width: 100% !important; /* Принудительно делаем адаптивными */
  }
}

@media (max-width: 480px) {
  .заголовок h1 {
    font-size: 1.5rem;
  }

  .кнопка-управления {
    display: block;
    width: 100%;
    margin: 10px 0;
  }
}
```

```
.эмулятор-устройств {
    flex-direction: column;
    align-items: center;
}

.устройство {
    width: 100%;
    max-width: 300px;
}
}

</style>
</head>
<body>
<div class="контейнер">
    <!-- ЗАГОЛОВОК -->
    <div class="заголовок">
        <h1>✨ max-width: 100% — Волшебство адаптивных изображений ✨</h1>
        <p>Научитесь делать картинки, которые красиво смотрятся на любом устройстве!</p>
    </div>

    <!-- ОСНОВНОЕ СОДЕРЖАНИЕ -->
    <div class="содержание">
        <div class="инфо-блок">
            <h3>⌚ Что такое адаптивное изображение?</h3>
            <p>Адаптивное изображение — это картинка, которая автоматически подстраивается под размер экрана устройства. На большом мониторе она большая, на маленьком телефоне — маленькая, но всегда остаётся в границах экрана и сохраняет пропорции.</p>
        </div>

        <!-- ЭКСПЕРИМЕНТ 1: ПРОБЛЕМА БЕЗ max-width: 100% -->
        <div class="эксперимент">
            <div class="эксперимент-заголовок">
                <div class="номер-эксперимента">1</div>
                <h2>Проблема: Фиксированное изображение</h2>
            </div>

            <div class="код-блок">
<span class="css-комментарий">/* ПЛОХОЙ ПРИМЕР - не адаптируется */</span><br>
<span class="css-селектор">img</span> {<br>
    width: 800px;
<span class="css-комментарий">/* Фиксированная ширина */</span><br>
}
</div>
```

<div class="предупреждение">  
    <strong>⚠ Проблема:</strong> Если экран меньше 800px, изображение выйдет за его границы. Появятся горизонтальные полосы прокрутки!  
    </div>

<h3>Демонстрация проблемы:</h3>  
<p>Измените размер окна браузера или выберите устройство на панели управления:</p>

<div class="эмюлятор-устройств">  
    <div class="устройство смартфон">  
        <div class="название-устройства">📱 Смартфон (300px)</div>  
        <div style="width: 300px; border: 2px dashed #3498db; padding: 10px; overflow: auto;">  
              
                <img alt="Картина не помещается!" x="100" y="100" width="200" height="200" />  

Нужно прокручивать вправо

  
        </div>  
    </div>  
</div>

<div class="устройство планшет">  
    <div class="название-устройства">💻 Планшет (500px)</div>  
    <div style="width: 500px; border: 2px dashed #9b59b6; padding: 10px; overflow: auto;">  
          
        <img alt="Всё ещё не помещается!" x="100" y="100" width="200" height="200" />  

Всё ещё нужно прокручивать

  
    </div>  
</div>

```

<!-- ЭКСПЕРИМЕНТ 2: РЕШЕНИЕ С max-width: 100% -->


<div class="эксперимент-заголовок">
        <div class="номер-эксперимента">2</div>
        <h2>Решение: max-width: 100%</h2>
    </div>

    <div class="код-блок">
/* ХОРОШИЙ ПРИМЕР - адаптируется */
img {  

    &nbsp;&nbsp;<span class="css-свойство">max-width</span>: <span class="css-значение">100%</span>; /* Максимум 100% от родителя */  

    &nbsp;&nbsp;<span class="css-свойство">height</span>: <span class="css-значение">auto</span>;  

/* Автоматическая высота */<br>
}
</div>

<div class="успех">
    <strong>✓ Решение:</strong> Изображение никогда не будет шире своего контейнера.  

На больших экранах будет большим, на маленьких — уменьшится, но всегда останется внутри границ!
</div>

<h3>Волшебство в действии:</h3>
<p>Посмотрите, как одно и то же изображение ведёт себя в контейнерах разного размера:</p>

<div class="сетка-изображений">
    <div class="изображение-блок">
        <h4>Маленький контейнер (300px)</h4>
        <div class="контейнер-малый">
            
        </div>
        <p style="color: #2ecc71; text-align: center; margin-top: 10px;">
            ✓ Идеально поместились!<br>
            Размер: 300px (100% от контейнера)
        </p>
    </div>
</div>


```

```

<div class="изображение-блок">
    <h4>Средний контейнер (500px)</h4>
    <div class="контейнер-средний">
        
    </div>
    <p style="color: #2ecc71; text-align: center; margin-top: 10px;">
        ✓ Идеально поместились!
        Размер: 500px (100% от контейнера)
    </p>
</div>

<div class="изображение-блок">
    <h4>Большой контейнер (800px)</h4>
    <div class="контейнер-большой">
        
    </div>
    <p style="color: #2ecc71; text-align: center; margin-top: 10px;">
        ✓ Идеально поместились!
        Размер: 800px (оригинальный размер)
    </p>
</div>

<div class="инфо-блок">
    <strong>Как это работает:</strong>
    <p><code>max-width: 100%</code> означает: «Ты можешь быть любого размера, но не больше 100% ширины твоего родительского контейнера».</p>
    <p><code>height: auto</code> означает: «Высота автоматически рассчитывается, чтобы сохранить пропорции изображения».</p>
</div>

```

<!-- ЭКСПЕРИМЕНТ 3: РАЗНЫЕ СЦЕНАРИИ -->

```
<div class="эксперимент">
  <div class="эксперимент-заголовок">
    <div class="номер-эксперимента">3</div>
    <h2>Разные сценарии использования</h2>
  </div>

  <h3>Сценарий 1: Галерея фотографий</h3>
  <div class="код-блок">
<span class="css-комментарий">/* Адаптивная галерея */</span><br>
<span class="css-селектор">.галерея img</span> {<br>
  &nbsp;&nbsp;<span class="css-свойство">max-width</span>: <span class="css-значение">100%</span>;<br>
  &nbsp;&nbsp;<span class="css-свойство">height</span>: <span class="css-значение">auto</span>;<br>
  &nbsp;&nbsp;<span class="css-свойство">display</span>: <span class="css-значение">block</span>;
<span class="css-комментарий">/* Убирает лишние отступы */</span><br>
  &nbsp;&nbsp;<span class="css-свойство">border-radius</span>: <span class="css-значение">10px</span>; <span class="css-комментарий">/* Скругленные углы */</span><br>
}<br>
</div>

<div class="пример-галерея">
  <div class="галерея-фото">
    
    <div class="подпись-галереи">Красные розы в саду</div>
  </div>

  <div class="галерея-фото">
    
    <div class="подпись-галереи">Весенние тюльпаны</div>
  </div>

  <div class="галерея-фото">
    
    <div class="подпись-галереи">Поля лаванды</div>
```

```
</div>

<div class="галерея-фото">
    
    <div class="подпись-галереи">Пушистые пионы</div>
</div>
</div>

<h3>Сценарий 2: Картинка с текстом</h3>
<div class="код-блок">
/* Статья с иллюстрацией */
<span class="css-селектор".статья img> {<br>
    max-width: 100%;<br>
    height: auto;<br>
    float: left;<br>
    /* Обтекание текстом */<br>
    margin: 0 20px 20px 0;<br>
    border: 3px solid #d2b48c;<br>
    border-radius: 10px;<br>
}
</div>

<div style="background: #f9f5f0; padding: 25px; border-radius: 10px; margin: 25px 0;">
    
<h4 style="margin-top: 0; color: #8b4513;">Мой прекрасный сад</h4>
<p>Каждое утро я начинаю с прогулки по своему саду. Здесь растут розы, тюльпаны, пионы и много других прекрасных цветов. Благодаря <strong>max-width: 100%</strong>, эта фотография моего сада будет красиво смотреться на любом устройстве.</p>
```

<p>На компьютере вы увидите её достаточно большой, а на телефоне она автоматически уменьшится, но останется чёткой и красивой. Никаких горизонтальных полос прокрутки!</p>

<p>Попробуйте изменить размер окна браузера или откройте эту страницу на телефоне — изображение адаптируется!</p>

</div>

<h3>Сценарий 3: Высокие изображения (портретные)</h3>

<div class="код-блок">

```
<span class="css-комментарий">/* Портретные фото с фиксированной высотой */</span><br>
<span class="css-селектор">.портретное-фото</span> {<br>
  &nbsp;&nbsp;<span class="css-свойство">max-width</span>: <span class="css-значение">100%</span>;<br>
  &nbsp;&nbsp;<span class="css-свойство">height</span>: <span class="css-значение">400px</span>;
  <span class="css-комментарий">/* Фиксированная высота */</span><br>
  &nbsp;&nbsp;<span class="css-свойство">object-fit</span>: <span class="css-значение">cover</span>;
  <span class="css-комментарий">/* Важно! Обрезает, но не искажает */</span><br>
  &nbsp;&nbsp;<span class="css-свойство">border-radius</span>: <span class="css-значение">10px</span>;<br>
}<br>
</div>
```

<div class="предупреждение">

<strong>⚠ Внимание с фиксированной высотой!</strong>

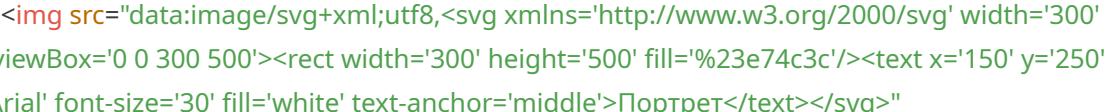
<p>Если задать и <code>max-width: 100%</code> и фиксированную <code>height</code>, изображение может искажиться. Используйте <code>object-fit: cover</code> или <code>object-fit: contain</code> чтобы решить эту проблему.</p>

</div>

<div style="display: flex; gap: 20px; flex-wrap: wrap; margin: 25px 0;">

<div style="flex: 1; min-width: 250px;">

<h4>Без object-fit (плохо):</h4>





alt="Портретное фото"

style="max-width: 100%; height: 200px; border: 3px solid #e74c3c; border-radius: 10px;">

<p style="color: #e74c3c; text-align: center;">Искажено! Сжато по высоте</p>

</div>

<div style="flex: 1; min-width: 250px;">

<h4>С object-fit: cover (хорошо):</h4>

```

<p style="color: #2ecc71; text-align: center;">Обрезано, но пропорции сохранены</p>
</div>
</div>
</div>

<!-- ПАНЕЛЬ УПРАВЛЕНИЯ -->
<div class="панель-управления">
<h3 style="color: white; margin-top: 0;">🎮 Лаборатория: Поэкспериментируйте!</h3>
<p style="color: #ecf0f1; margin-bottom: 25px;">Измените размер окна браузера или выберите режим эмуляции устройства:</p>

<button class="кнопка-управления" onclick="эмулироватьУстройство('mobile')">📱 Эмуляция смартфона</button>
<button class="кнопка-управления" onclick="эмулироватьУстройство('tablet')">💻 Эмуляция планшета</button>
<button class="кнопка-управления" onclick="эмулироватьУстройство('desktop')">🖥️ Эмуляция компьютера</button>
<button class="кнопка-управления" onclick="сброситьЭмуляцию()" style="background: #e74c3c;">🔄 Сбросить эмуляцию</button>

<div id="статусЭмуляции" style="margin-top: 20px; padding: 15px; background: rgba(255,255,255,0.1); border-radius: 8px; display: none;">
<strong>Режим эмуляции:</strong> <span id="текущееУстройство"></span>
<p style="margin: 10px 0 0 0; font-size: 0.9rem;">Измените размер окна браузера, чтобы увидеть как адаптируются изображения</p>
</div>
</div>

<!-- ПРАКТИЧЕСКИЕ СОВЕТЫ -->
<div class="эксперимент">
<div class="эксперимент-заголовок">
<div class="номер-эксперимента">4</div>
<h2>Практические советы и готовые решения</h2>
</div>

<h3>Готовый CSS для всех изображений на сайте:</h3>
<div class="код-блок">
<span class="css-комментарий">/* Добавьте этот код в начало вашего файла style.css */</span><br>
```

```
<span class="css-селектор">img</span> {<br>
  &nbsp;&nbsp;<span class="css-свойство">max-width</span>: <span class="css-
  значение">100%</span>;<br>
  &nbsp;&nbsp;<span class="css-свойство">height</span>: <span class="css-
  значение">auto</span>;<br>
  &nbsp;&nbsp;<span class="css-свойство">display</span>: <span class="css-значение">block</span>;
  <span class="css-комментарий">/* Убирает лишние отступы снизу */</span><br>
}<br><br>
<span class="css-комментарий">/* Для портретных фото с фиксированной высотой */</span><br>
<span class="css-селектор">.портрет</span> {<br>
  &nbsp;&nbsp;<span class="css-свойство">object-fit</span>: <span class="css-
  значение">cover</span>;<br>
  &nbsp;&nbsp;<span class="css-свойство">width</span>: <span class="css-
  значение">100%</span>;<br>
  &nbsp;&nbsp;<span class="css-свойство">height</span>: <span class="css-значение">300px</span>;
  <span class="css-комментарий">/* Или любая другая высота */</span><br>
}
</div>
```

<h3>Что делать, если изображение слишком маленькое на больших экранах?</h3>

```
<div class="код-блок">
<span class="css-комментарий">/* Ограничение минимального размера */</span><br>
<span class="css-селектор">.важное-изображение</span> {<br>
  &nbsp;&nbsp;<span class="css-свойство">max-width</span>: <span class="css-
  значение">100%</span>;<br>
  &nbsp;&nbsp;<span class="css-свойство">min-width</span>: <span class="css-
  значение">200px</span>; <span class="css-комментарий">/* Не меньше 200px */</span><br>
  &nbsp;&nbsp;<span class="css-свойство">height</span>: <span class="css-
  значение">auto</span>;<br>
}
</div>
```

<h3>Адаптивное изображение с рамкой и тенью:</h3>

```
<div class="код-блок">
<span class="css-комментарий">/* Красивое адаптивное изображение */</span><br>
<span class="css-селектор">.красивое-фото</span> {<br>
  &nbsp;&nbsp;<span class="css-свойство">max-width</span>: <span class="css-
  значение">100%</span>;<br>
  &nbsp;&nbsp;<span class="css-свойство">height</span>: <span class="css-
  значение">auto</span>;<br>
  &nbsp;&nbsp;<span class="css-свойство">border</span>: <span class="css-значение">10px solid
  white</span>;<br>
  &nbsp;&nbsp;<span class="css-свойство">border-radius</span>: <span class="css-
  значение">15px</span>;<br>
```

```
&nbsp;&nbsp;<span class="css-свойство">box-shadow</span>: <span class="css-значение">0 10px  
30px rgba(0,0,0,0.2)</span>;<br>  
&nbsp;&nbsp;<span class="css-свойство">display</span>: <span class="css-  
значение">block</span>;<br>  
&nbsp;&nbsp;<span class="css-свойство">margin</span>: <span class="css-значение">0 auto</span>;  
<span class="css-комментарий">/* Центрирование */</span><br>  
>  
</div>  
  
<div style="text-align: center; margin: 30px 0;">  
      
    <p style="color: #666; margin-top: 15px; font-style: italic;">Это изображение адаптируется  
под любой экран, но всегда выглядит красиво!</p>  
    </div>  
    </div>  
</div>  
  
<!-- ИТОГ -->  
<div class="итог">  
    <h2 style="color: white;">🎉 Поздравляю! Теперь ваши изображения адаптивны!</h2>  
  
<div class="чекелист">  
    <div class="чекелист-пункт">  
        <h4>Вы научились:</h4>  
        <p>Использовать <code>max-width: 100%</code> для адаптивных изображений</p>  
    </div>  
  
    <div class="чекелист-пункт">  
        <h4>Вы понимаете:</h4>  
        <p>Почему <code>height: auto</code> сохраняет пропорции картинок</p>  
    </div>  
  
    <div class="чекелист-пункт">  
        <h4>Вы можете:</h4>  
        <p>Создавать галереи, которые работают на любых устройствах</p>  
    </div>
```

```
<div class="чекалист-пункт">
  <h4>Вы знаете:</h4>
  <p>Как использовать <code>object-fit</code> для портретных фотографий</p>
</div>
</div>

<div style="margin-top: 40px;">
  <h3 style="color: #2ecc71;">Ваше домашнее задание:</h3>
  <p style="font-size: 1.2rem; max-width: 800px; margin: 20px auto;">
    1. Добавьте <code>max-width: 100%; height: auto;</code> ко всем изображениям на
    вашем сайте<br>
    2. Создайте адаптивную галерею из 4-6 фотографий вашего хобби<br>
    3. Протестируйте на разных размерах экрана (изменяйте размер окна браузера)
  </p>
  <button class="кнопка-управления" onclick="alert('Удачи в создании адаптивных
изображений! Помните: один раз добавив max-width: 100%, вы решаете проблему для всех
устройств! 📱💻💻')" style="background: #2ecc71; margin-top: 20px;">
    Принять задание!
  </button>
</div>
</div>
</div>

<script>
// ФУНКЦИИ ДЛЯ ЭМУЛИАЦИИ УСТРОЙСТВ
function эмулироватьУстройство(тип) {
  const статус = document.getElementById('статусЭмуляции');
  const устройство = document.getElementById('текущееУстройство');

  switch(тип) {
    case 'mobile':
      устройство.textContent = 'Смартфон (ширина 300-500px)';
      alert('Измените размер окна браузера примерно до 400px в ширину, чтобы увидеть, как
адаптируются изображения на смартфоне!');
      break;

    case 'tablet':
      устройство.textContent = 'Планшет (ширина 600-800px)';
      alert('Измените размер окна браузера примерно до 700px в ширину, чтобы увидеть, как
адаптируются изображения на планшете!');
      break;

    case 'desktop':
      устройство.textContent = 'Компьютер (ширина 1000px+);
```

```
    alert('Растяните окно браузера на весь экран, чтобы увидеть, как адаптируются изображения на компьютере!');
    break;
}

статус.style.display = 'block';
}

function сброситьЭмуляцию() {
    document.getElementById('статусЭмуляции').style.display = 'none';
    alert('Эмуляция сброшена. Теперь вы можете свободно изменять размер окна браузера.');
}

// ФУНКЦИЯ ДЛЯ ИНТЕРАКТИВНОЙ ДЕМОНСТРАЦИИ
document.addEventListener('DOMContentLoaded', function() {
    // Добавляем обработчик изменения размера окна
    window.addEventListener('resize', function() {
        const ширина = window.innerWidth;
        const статус = document.getElementById('статусЭмуляции');

        if (статус.style.display === 'block') {
            const устройство = document.getElementById('текущееУстройство');
            let тип = "";

            if (ширина < 600) {
                тип = '(сейчас: смартфон)';
            } else if (ширина < 1000) {
                тип = '(сейчас: планшет)';
            } else {
                тип = '(сейчас: компьютер)';
            }

            // Обновляем текст, сохраняя исходный тип устройства
            const исходныйТекст = устройство.textContent.replace(/\(сейчас:.*\)/, "");
            устройство.textContent = исходныйТекст + тип;
        }
    });
});

// Добавляем подсказки для изображений
document.querySelectorAll('img').forEach(img => {
    img.addEventListener('click', function() {
        const ширина = this.naturalWidth || this.width;
        const высота = this.naturalHeight || this.height;
        const стили = window.getComputedStyle(this);

```

```

    alert(`Информация об изображении:\n\n` +
      `Размер файла: ${ширина}x${высота} пикселей\n` +
      `Отображается: ${Math.round(this.width)}x${Math.round(this.height)}px\n` +
      `max-width: ${стили.maxWidth}\n` +
      `height: ${стили.height}\n\n` +
      `Это изображение адаптируется под размер экрана благодаря max-width: 100%`);

  });

}

// Добавляем эффект при наведении на галерею
document.querySelectorAll('.галерея-фото').forEach(фото => {
  фото.addEventListener('mouseenter', function() {
    this.style.transform = 'scale(1.05)';
    this.style.boxShadow = '0 15px 40px rgba(0,0,0,0.2)';
  });

  фото.addEventListener('mouseleave', function() {
    this.style.transform = 'scale(1)';
    this.style.boxShadow = '0 5px 15px rgba(0,0,0,0.1)';
  });

});

// ФУНКЦИЯ ДЛЯ ТЕСТИРОВАНИЯ НА РАЗНЫХ УСТРОЙСТВАХ
function протестироватьНаУстройстве() {
  const устройства = [
    { название: 'iPhone SE', ширина: 375, высота: 667 },
    { название: 'iPad', ширина: 768, высота: 1024 },
    { название: 'Ноутбук', ширина: 1366, высота: 768 },
    { название: 'Монитор', ширина: 1920, высота: 1080 }
  ];

  let сообщение = 'Рекомендуемые размеры для тестирования:\n\n';
  устройства.forEach(устройство => {
    сообщение += `${устройство.название}: ${устройство.ширина}x${устройство.высота}px\n`;
  });

  сообщение += '\nИзмените размер окна браузера до этих значений и посмотрите, как
адаптируются изображения!';

  alert(сообщение);
}

```

```
// Добавляем кнопку тестирования, если её нет
document.addEventListener('DOMContentLoaded', function() {
  const панель = document.querySelector('.панель-управления');
  if (панель) {
    const кнопка = document.createElement('button');
    кнопка.className = 'кнопка-управления';
    кнопка.textContent = '📱 Тестовые размеры';
    кнопка.onclick = протестироватьНаУстройстве;
    кнопка.style.background = '#9b59b6';
    панель.appendChild(кнопка);
  }
});
</script>
</body>
</html>
```

---

## ЧАСТЬ 3: Основные принципы работы max-width: 100%

### Что делает эта магия?

css

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

### Пошаговое объяснение:

1. max-width: 100% — говорит изображению:

"Ты можешь быть любого размера, но НЕ БОЛЬШЕ, чем 100% ширины твоего родительского контейнера"

2. height: auto — говорит изображению:

"Рассчитай высоту автоматически, чтобы сохранить пропорции (не искажайся)!"

### Примеры:

Если родительский контейнер 300px:

- Изображение 800px → сожмётся до 300px
- Изображение 200px → останется 200px (не растянется!)

**Если родительский контейнер 800px:**

- Изображение 800px → останется 800px
  - Изображение 1200px → сожмётся до 800px
  - Изображение 400px → останется 400px
- 

## ЧАСТЬ 4: Разница между `width: 100%` и `max-width: 100%`

 **width: 100% — ПЛОХО для адаптивности:**

css

```
img {  
    width: 100%; /* Всегда растягивается на 100% */  
}
```

**Проблема:** Маленькое изображение 200px растягивается до 800px на большом экране → будет размытым и некачественным!

 **max-width: 100% — ХОРОШО для адаптивности:**

css

```
img {  
    max-width: 100%; /* Не больше 100%, но может быть меньше */  
    height: auto; /* Сохраняем пропорции */  
}
```

**Преимущество:** Маленькое изображение 200px останется 200px даже на большом экране → останется чётким!

---

## ЧАСТЬ 5: Готовые решения для разных ситуаций

### Решение 1: Базовая настройка для ВСЕХ изображений

css

```
/* Добавьте в начало вашего CSS-файла */  
img {  
    max-width: 100%;  
    height: auto;  
    display: block; /* Убирает лишние отступы снизу */  
}
```

### Решение 2: Адаптивная галерея

css

```
.gallery {  
    display: grid;  
    grid-template-columns: repeat(auto-fill, minmax(250px, 1fr));  
    gap: 20px;  
}  
  
.gallery img {  
    width: 100%;  
    height: 200px;  
    object-fit: cover; /* Обрезает, но не искажает */  
    border-radius: 10px;  
    transition: transform 0.3s;  
}  
  
.gallery img:hover {  
    transform: scale(1.05);  
}
```

### Решение 3: Изображение с подписью

css

```
.figure {  
    max-width: 100%;  
    margin: 20px 0;  
}
```

```
.figure img {  
    max-width: 100%;  
    height: auto;  
    border: 5px solid #f0f0f0;  
    border-radius: 10px;  
}  
  
.caption {  
    text-align: center;  
    font-style: italic;  
    color: #666;  
    margin-top: 10px;  
}
```

## Решение 4: Фоновая картинка (для div)

css

```
.hero {  
    width: 100%;  
    height: 400px;  
    background-image: url('ваша-картинка.jpg');  
    background-size: cover; /* Заполняет контейнер */  
    background-position: center; /* Центрирует */  
    background-repeat: no-repeat; /* Не повторяет */  
}
```

---

## ЧАСТЬ 6: Частые ошибки и их решение

### ✖ Ошибка 1: Только max-width без height: auto

css

```
img {  
    max-width: 100%;  
    /* height не задан */  
}
```

**Проблема:** Изображение может исказиться

**Решение:** Всегда добавляйте height: auto

## Ошибка 2: width: 100% и max-width: 100% вместе

css

```
img {  
    width: 100%; /* Переопределяет max-width! */  
    max-width: 100%;  
}
```

**Проблема:** width: 100% побеждает, изображение всегда растягивается

**Решение:** Используйте только max-width: 100%

## Ошибка 3: Фиксированная высота без object-fit

css

```
img {  
    max-width: 100%;  
    height: 300px; /* Фиксированная высота */  
}
```

**Проблема:** Портретные фото искажаются

**Решение:** Добавьте object-fit: cover или object-fit: contain

## Правильное решение:

css

```
img {  
    max-width: 100%;  
    height: auto; /* Для обычных фото */  
}  
  
.portrait {  
    max-width: 100%;  
    height: 300px; /* Для фиксированной высоты */  
    object-fit: cover; /* Обрезает, но не искажает */  
}
```

---

## ЧАСТЬ 7: Тестирование адаптивности

### Как проверить, что ваши изображения адаптивны:

1. Изменяйте размер окна браузера — тяните за угол
2. Используйте инструменты разработчика (F12):

- Нажмите Ctrl+Shift+M (или Cmd+Option+M на Mac)
- Выберите разные устройства из списка
- Проверьте, не выходят ли изображения за границы

3. Откройте сайт на реальных устройствах:

- Смартфоне
- Планшете
- Компьютере с разными мониторами

### Признаки правильно адаптивных изображений:

- ✓ Нет горизонтальной прокрутки
  - ✓ Изображения не размыты
  - ✓ Пропорции сохранены
  - ✓ На всех устройствах смотрятся хорошо
- 

## ЧАСТЬ 8: Практическое задание

### Задание «Создай адаптивную галерею своего хобби»

1. Создайте HTML-строктуру:

html

```
<div class="gallery">
  <div class="photo">
    
    <p class="caption">Подпись к фото 1</p>
  </div>
  <div class="photo">
    
    <p class="caption">Подпись к фото 2</p>
```

```
</div>
<!-- Добавьте ещё 2-4 фото -->
</div>
2. Добавьте CSS:
css

/* Сделайте все изображения адаптивными */
img {
    max-width: 100%;
    height: auto;
    display: block;
}

/* Создайте сетку для галереи */
.gallery {
    display: grid;
    grid-template-columns: repeat(auto-fill, minmax(250px, 1fr));
    gap: 20px;
    margin: 30px 0;
}

/* Стили для каждой фотографии */
.photo {
    border: 1px solid #ddd;
    border-radius: 10px;
    overflow: hidden;
    transition: transform 0.3s;
}

.photo:hover {
    transform: translateY(-5px);
    box-shadow: 0 10px 20px rgba(0,0,0,0.1);
}

/* Подпись к фото */
.caption {
    padding: 15px;
    text-align: center;
    background: #f9f9f9;
    margin: 0;
}
```

### 3. Протестируйте:

- Измените размер окна браузера
- Откройте на телефоне (если есть возможность)

■ Проверьте в инструментах разработчика

---

## ИТОГ:

Теперь вы владеете одной из самых важных магий веб-дизайна! Вы научились:

1. **Делать изображения адаптивными** с помощью `max-width: 100%`
2. **Сохранять пропорции** с помощью `height: auto`
3. **Работать с разными типами изображений** (пейзажные, портретные)
4. **Использовать** `object-fit` для фиксированных высот
5. **Создавать адаптивные галереи** и другие композиции

**Золотое правило:** Всегда добавляйте `max-width: 100%; height: auto;` к изображениям на ваших сайтах. Эта простая строка кода решает множество проблем с отображением на разных устройствах!

## Формула успеха:

text

`max-width: 100% + height: auto = ✓ Адаптивное изображение`

Теперь ваши сайты будут красиво выглядеть на любом устройстве — от смартфона до большого монитора! В следующей главе мы научимся добавлять рамки и скруглять углы у изображений, чтобы они выглядели ещё более профессионально и современно.

## ■ Простая рамка и скругление углов (border-radius) – чтобы фото выглядело современно.

Дорогой друг, представьте, что у вас есть любимая фотография в альбоме. Вы можете просто вклейть её на страницу, а можно сделать красивее — вставить в элегантную рамку или даже обрезать углы, чтобы фотография выглядела мягче и современнее. В веб-дизайне мы делаем то же самое с помощью двух волшебных свойств: **border** (рамка) и **border-radius** (скругление углов). Давайте научимся применять их к изображениям!

---

### 1. Рамка для фотографии — свойство border

Рамка — это как паспарту для картины. Она отделяет изображение от фона, придаёт ему законченный вид и может стать стильным акцентом.

**Базовая рамка:**

css

```
.foto-rama {  
    border: 3px solid #8b4513; /* Рамка: толщина 3px, сплошная, коричневого цвета */  
}
```

**Что здесь написано:**

- Зрх — толщина рамки (можете ставить 1px, 2px, 5px, 10px — как вам нравится!)
- solid — стиль линии (сплошная). Ещё бывают:
  - dotted — точечная (.....)
  - dashed — пунктирная (---)
  - double — двойная
- #8b4513 — цвет рамки (можно использовать любые цвета, которые мы уже изучили!)

**Пример с разными рамками:**

css

```
.foto-rama1 {  
    border: 5px solid #3498db; /* Синяя сплошная рамка */
```

```
}

.foto-rama2 {
    border: 2px dashed #e74c3c; /* Красная пунктирная рамка */
}

.foto-rama3 {
    border: 8px double #2ecc71; /* Зелёная двойная рамка */
}
```

---

## 2. Скругление углов — свойство `border-radius`

Современный дизайн часто использует скруглённые углы — они выглядят мягче, дружелюбнее и современнее, чем острые.

### Простое скругление всех углов:

css

```
.foto-skruglennaya {
    border-radius: 10px; /* Все 4 угла скруглены на 10 пикселей */
}
```

### Что здесь происходит:

Чем больше число, тем сильнее скругление:

- `border-radius: 5px;` — лёгкое скругление
- `border-radius: 15px;` — заметное скругление
- `border-radius: 25px;` — сильное скругление
- `border-radius: 50%;` — **магия!** Фотография становится круглой!

### Разное скругление для разных углов:

Можно управлять каждым углом отдельно:

css

```
.foto-neobychnaya {
    border-radius: 15px 5px 20px 10px;
    /* Верхний-левый = 15px, верхний-правый = 5px,
       нижний-левый = 20px, нижний-правый = 10px */
}
```

```
        нижний-правый = 20px, нижний-левый = 10px */
    }
```

**Запоминаем по часовой стрелке:** верхний-левый → верхний-правый → нижний-правый → нижний-левый.

---

### 3. Практикум: Создаём фотогалерею с красивыми рамками

Давайте создадим страницу, где применим всё, что узнали! Создайте файл foto-galereya.html:

html

```
<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Моя фото-галерея с рамками</title>
    <style>
        /* Общие настройки */
        body {
            font-family: 'Georgia', serif;
            background-color: #f9f5f0; /* Кремовый фон */
            padding: 30px;
            max-width: 1000px;
            margin: 0 auto;
            line-height: 1.6;
        }

        h1 {
            text-align: center;
            color: #8b4513; /* Коричневый */
            margin-bottom: 40px;
            border-bottom: 3px solid #d2b48c;
            padding-bottom: 15px;
        }

        /* Кон테йнер для галереи */
        .galereya {
            display: flex;
```

```
flex-wrap: wrap;
gap: 30px;
justify-content: center;
}

/* Общий стиль для всех фото-блоков */
.foto-block {
    width: 300px;
    text-align: center;
    padding: 20px;
    background: white;
    border-radius: 15px;
    box-shadow: 0 10px 20px rgba(0,0,0,0.1);
}

/* Самы изображения */
.foto-block img {
    width: 100%; /* Фото занимает всю ширину блока */
    height: 200px;
    object-fit: cover; /* Обрезаем фото, чтобы оно красиво вписывалось */
    display: block; /* Убираем лишние отступы */
}

/* ПОДПИСЬ под фото */
.podpis {
    margin-top: 15px;
    font-style: italic;
    color: #666;
    font-size: 16px;
}

/* --- РАЗНЫЕ ВИДЫ РАМОК И СКРУГЛЕНИЙ --- */

/* 1. Классическая деревянная рамка */
.rama-klassicheskaya img {
    border: 8px solid #8b4513; /* Толстая коричневая рамка */
    border-radius: 5px; /* Слегка скруглённые углы */
}

/* 2. Современная круглая фотография */
.rama-kruglaya img {
    border: 4px solid #3498db; /* Тонкая синяя рамка */
    border-radius: 50%; /* МАГИЯ! Делаем фото круглым! */
    padding: 5px; /* Отступ между фото и рамкой */
}
```

```
background-color: white; /* Фон под рамкой */
}

/* 3. Мягкая рамка с тенью */
.rama-myagkaya img {
    border: 2px solid #e6d5c3; /* Светлая рамка */
    border-radius: 20px; /* Сильное скругление */
    box-shadow: 0 5px 15px rgba(0,0,0,0.1); /* Лёгкая тень */
}

/* 4. Пунктирная винтажная рамка */
.rama-vintazhnaya img {
    border: 6px dashed #a0522d; /* Коричневый пунктири */
    border-radius: 10px 30px 10px 30px; /* Разные углы для узора */
    padding: 8px;
    background-color: #ffffaf5;
}

/* 5. Без рамки, только скругление */
.rama-bezramki img {
    border-radius: 15px; /* Скругление без рамки */
    box-shadow: 0 8px 20px rgba(0,0,0,0.15); /* Тень вместо рамки */
}

/* 6. Двойная рамка */
.rama-dvoinaya img {
    border: 5px double #2ecc71; /* Двойная зелёная рамка */
    border-radius: 0; /* Острые углы — для контраста */
    padding: 10px;
    background-color: #f0ffff;
}

/* Информационная панель */
.info-panel {
    background: #fff8e1;
    padding: 25px;
    border-radius: 15px;
    margin-top: 40px;
    border-left: 6px solid #ffb300;
}

/* Кнопка для экспериментов */
.knopka {
    display: inline-block;
```

```
background: #8b4513;
color: white;
padding: 12px 24px;
border-radius: 8px;
text-decoration: none;
font-weight: bold;
margin-top: 15px;
border: none;
cursor: pointer;
}

.knopka:hover {
background: #a0522d;
}

</style>
</head>
<body>
<h1>Моя фото-галерея: эксперименты с рамками</h1>

<div class="galereya">
<!-- Фото 1: Классическая рамка -->
<div class="foto-block rama-klassicheskaya">

<p class="podpis">Классическая деревянная рамка</p>
<p><small>border: 8px solid #8b4513;<br>border-radius: 5px;</small></p>
</div>

<!-- Фото 2: Круглая -->
<div class="foto-block rama-kruglaya">

<p class="podpis">Современная круглая фотография</p>
<p><small>border-radius: 50%;<br>border: 4px solid #3498db;</small></p>
</div>

<!-- Фото 3: Мягкая рамка -->
<div class="foto-block rama-myagkaya">

<p class="podpis">Мягкая рамка с тенью</p>
<p><small>border-radius: 20px;<br>box-shadow для объёма</small></p>
</div>
```

```
<!-- Фото 4: Винтажная -->
<div class="foto-block rama-vintazhnaya">
    
        <p class="podpis">Пунктирная винтажная рамка</p>
        <p><small>border: 6px dashed #a0522d;<br>разные углы скругления</small></p>
</div>
```

```
<!-- Фото 5: Без рамки -->
<div class="foto-block rama-bezramki">
    
        <p class="podpis">Только скругление и тень</p>
        <p><small>border-radius: 15px;<br>без рамки, только тень</small></p>
</div>
```

```
<!-- Фото 6: Двойная рамка -->
<div class="foto-block rama-dvoynaya">
    
        <p class="podpis">Двойная рамка с острыми углами</p>
        <p><small>border: 5px double #2ecc71;<br>border-radius: 0;</small></p>
</div>
</div>
```

```
<div class="info-panel">
    <h2>Как это работает?</h2>
    <p><strong>border</strong> — создаёт рамку вокруг элемента. Состоит из трёх
частей:</p>
    <ul>
        <li><strong>Толщина:</strong> 1px, 3px, 5px, 10px — чем больше, тем толще
рамка</li>
        <li><strong>Стиль:</strong> solid (сплошная), dashed (пунктир), dotted (точки), double
(двойная)</li>
        <li><strong>Цвет:</strong> любой цвет в любом формате (#FF0000, red,
rgb(255,0,0))</li>
    </ul>
```

```
<p><strong>border-radius</strong> — скругляет углы элемента:</p>
<ul>
    <li><strong>Одно значение:</strong> border-radius: 10px — все углы одинаковые</li>
    <li><strong>Два значения:</strong> border-radius: 10px 20px — верх-левый+ни-
правый=10px, верх-правый+ниж-левый=20px</li>
```

```

<li><strong>Четыре значения:</strong> border-radius: 5px 10px 15px 20px — каждый
угол свой (по часовой стрелке)</li>
<li><strong>50%:</strong> делает элемент круглым или овальным!</li>
</ul>

<h3>Эксперимент:</h3>
<p>Попробуйте изменить значения в CSS-коде и посмотрите, как меняются рамки!
Например:</p>
<ul>
    <li>Измените <code>border-radius: 50%;</code> на <code>border-radius:<br/>30px;</code></li>
    <li>Поменяйте <code>solid</code> на <code>dotted</code> в любой рамке</li>
    <li>Сделайте все рамки одного цвета</li>
</ul>

<button class="knopka" onclick="alert('Откройте файл в редакторе и меняйте значения! Не
бойтесь экспериментировать!')">
    Начать эксперимент
</button>
</div>

<div style="text-align: center; margin-top: 40px; color: #666; font-style: italic;">
    <p>💡 <strong>Совет:</strong> Сочетайте рамки с отступами (padding) внутри блока,
чтобы фотография не прилипала к рамке!</p>
</div>
</body>
</html>

```

---

## 4. Полезные советы по использованию рамок и скруглений

### 1. Сочетайте с отступами (padding):

css

```
.foto-udobnaya {
    border: 5px solid #8b4513;
    border-radius: 15px;
    padding: 15px; /* Пространство между фото и рамкой */
    background-color: white; /* Фон в этом пространстве */
}
```

### 2. Используйте разные рамки для разных типов фото:

- ▣ **Портреты:** лёгкие скругления, тонкие рамки

- **Пейзажи:** классические толстые рамки
- **Арт-фото:** необычные пунктирные или двойные рамки
- **Семейные фото:** круглые или овальные рамки

### 3. Создавайте "фото-полароид":

css

```
.foto-polaroid {  
    border: 1px solid #ccc;  
    border-radius: 2px;  
    padding: 15px 15px 40px 15px; /* Больше снизу для подписи */  
    background: white;  
    box-shadow: 0 4px 8px rgba(0,0,0,0.2);  
}
```

### 4. Экспериментируйте с цветами рамок:

- Сочетайте с цветовой гаммой сайта
- Используйте контрастные цвета для акцента
- Попробуйте полупрозрачные рамки: `border: 5px solid rgba(139, 69, 19, 0.7);`

---

## 5. Домашнее задание

### Задание 1: Создайте свою фото-витрину

1. Создайте страницу с 4 своими любимыми фотографиями (можно использовать заглушки)
2. Для каждой придумайте свой стиль рамки
3. Добавьте подписи к фотографиям

### Задание 2: "До и после"

1. Создайте два блока с одной и той же фотографией
2. В первом оставьте фото без рамки
3. Во втором добавьте красивую рамку и скругление
4. Подпишите: "Без оформления" и "С современным оформлением"

### Задание 3: Эксперимент с круглыми фото

1. Сделайте 3 круглых фотографии разного диаметра
2. Добавьте им разные цветные рамки

3. Расположите в ряд (используйте `display: flex`)

**Бонус: Создайте рамку, которая меняется при наведении мыши:**

css

```
.foto-interaktivnaya {  
    border: 3px solid #3498db;  
    border-radius: 10px;  
    transition: all 0.3s; /* Плавное изменение */  
}  
  
.foto-interaktivnaya:hover {  
    border: 5px solid #e74c3c; /* Толще и краснее при наведении */  
    border-radius: 20px; /* Больше скругление */  
    transform: scale(1.05); /* Немного увеличивается */  
}
```

---

## 6. Важно запомнить

1. `border` — рамка: толщина, стиль, цвет
2. `border-radius` — скругление углов
3. `50%` для `border-radius` делает круг
4. Всегда можно комбинировать: рамка + скругление + тень + отступы
5. Экспериментируйте! Это безопасно — если не понравится, всегда можно вернуть обратно.

**Дорогой друг, теперь ваши фотографии на сайте будут выглядеть не просто как картинки, а как настоящие произведения искусства в красивых рамках! В следующей части мы научимся делать адаптивные изображения, которые красиво смотрятся на любом устройстве — от компьютера до телефона.**

## ■ Как добавить подпись к фотографии?

Дорогой друг, представьте, что вы пришли на выставку картин. Под каждой работой есть аккуратная табличка: название, автор, год создания. Эта табличка помогает понять, что перед нами. На сайте мы делаем то же самое — добавляем подписи к фотографиям. Это не только красиво, но и полезно: подпись объясняет, что на фото, помогает в поиске и делает страницу профессиональнее.

---

### 1. Самый простой способ — текст рядом с фото

**HTML:**

```
html
<div class="foto-s-podpis">
    
    <p class="podpis">Моя любимая роза "Королева сада". Май 2024 года.</p>
</div>
```

**CSS:**

```
css
.foto-s-podpis {
    text-align: center; /* Центрируем фото и подпись */
    margin: 20px 0;
}

.foto-s-podpis img {
    width: 300px; /* Фиксируем ширину */
    border-radius: 10px;
    box-shadow: 0 5px 15px rgba(0,0,0,0.1);
}

.podpis {
    font-style: italic;
    color: #666;
    margin-top: 10px;
    font-size: 14px;
}
```

## Что получится:

<https://%D0%BF%D1%83%D1%82%D1%8C-%D0%BA-%D1%84%D0%BE>

%D1%82%D0%BE

Моя любимая роза "Королева сада". Май 2024 года.

---

## 2. Подпись поверх фотографии (как в журнале)

Иногда подпись размещают прямо на фото, особенно если это галерея или портфолио.

### HTML:

```
html
<div class="foto-s-nakladnoj-podpis">
    
    <div class="podpis-nakladnaya">
        <p>Закат на нашей даче. Июнь 2024</p>
    </div>
</div>
```

### CSS:

```
css
.foto-s-nakladnoj-podpis {
    position: relative; /* Важно! Это "якорь" для подписи */
    display: inline-block; /* Чтобы подпись не растягивалась на всю ширину */
    margin: 20px;
}

.foto-s-nakladnoj-podpis img {
    width: 400px;
    display: block; /* Убираем лишние отступы */
}

.podpis-nakladnaya {
    position: absolute; /* Подпись будет "плавать" поверх фото */
    bottom: 0; /* Прижимаем к низу фото */
```

```
    left: 0;
    right: 0;
    background: rgba(0, 0, 0, 0.7); /* Полупрозрачный тёмный фон */
    color: white;
    padding: 10px 15px;
    font-size: 16px;
    text-align: center;
}


```

### Что получится:

Фотография с полупрозрачной тёмной полосой внизу, на которой белым текстом написана подпись.

---

## 3. Стилизованная подпись в рамке (как музейная табличка)

### HTML:

```
html

<div class="foto-vitrina">
  <div class="foto-container">
    
  </div>
  <div class="podpis-ramka">
    <h3>«Первая клубника»</h3>
    <p>Моя внучка Машенька собирает первый урожай клубники</p>
    <p class="data">Июнь 2024</p>
  </div>
</div>
```

### CSS:

```
css

.foto-vitrina {
  width: 350px;
  margin: 30px auto;
  background: white;
  border-radius: 15px;
  overflow: hidden; /* Чтобы углы фотографии не вылезали за рамку */
  box-shadow: 0 10px 30px rgba(0,0,0,0.15);
```

```
}

.foto-container img {
    width: 100%;
    height: 250px;
    object-fit: cover; /* Фото красиво заполняет область */
    display: block;
}

.podpis-ramka {
    padding: 20px;
    border-top: 3px solid #d2b48c; /* Декоративная линия */
    background: #f9f5f0;
}

.podpis-ramka h3 {
    color: #8b4513;
    margin-top: 0;
    font-size: 22px;
}

.podpis-ramka p {
    color: #666;
    line-height: 1.5;
    margin: 10px 0;
}

.podpis-ramka .data {
    font-style: italic;
    color: #999;
    text-align: right;
    font-size: 14px;
}
```

## Что получится:

Аккуратная карточка с фотографией и стилизованной подписью снизу, как в музее или галерее.

---

## 4. Подпись при наведении мыши (интерактивный вариант)

Идеально для фото-галерей — подпись появляется только когда наводишь курсор.

### HTML:

```
html  
<div class="foto-s-skrytoj-podpis">  
      
    <div class="skrytaya-podpis">  
        <p>Чёрное море, Анапа. Наш первый совместный отпуск с мужем.</p>  
        <p>Сентябрь 2023</p>  
    </div>  
</div>
```

### CSS:

```
css  
.foto-s-skrytoj-podpis {  
    position: relative;  
    display: inline-block;  
    margin: 15px;  
    cursor: pointer;  
}  
  
.foto-s-skrytoj-podpis img {  
    width: 300px;  
    height: 200px;  
    object-fit: cover;  
    border-radius: 8px;  
    transition: all 0.3s;  
}  
  
.foto-s-skrytoj-podpis:hover img {  
    filter: brightness(0.7); /* Затемняем фото при наведении */  
}  
  
.skrytaya-podpis {  
    position: absolute;  
    bottom: 0;  
    left: 0;
```

```
right: 0;  
background: rgba(0, 0, 0, 0.8);  
color: white;  
padding: 15px;  
transform: translateY(100%); /* Изначально скрыта под фото */  
transition: transform 0.3s;  
border-radius: 0 0 8px 8px;  
}  
  
.foto-s-skrytoj-podpis:hover .skrytaya-podpis {
```

```
    transform: translateY(0); /* Поднимаем наверх при наведении */  
}
```

**Что получится:** При наведении мыши на фото оно слегка темнеет, и снизу плавно выезжает подпись.

---

## 5. Полный пример: Фото-дневник с подписями

Давайте создадим целую страницу, как дневник садовода:

html

```
<!DOCTYPE html>  
<html lang="ru">  
<head>  
    <meta charset="UTF-8">  
    <meta name="viewport" content="width=device-width, initial-scale=1.0">  
    <title>Мой садовый дневник</title>  
    <style>  
        body {  
            font-family: 'Georgia', serif;  
            background-color: #f0fff0;  
            color: #333;  
            padding: 20px;  
            max-width: 1000px;  
            margin: 0 auto;  
        }  
  
        h1 {  
            text-align: center;  
            color: #2e8b57;
```

```
border-bottom: 3px solid #98fb98;
padding-bottom: 15px;
}

.dnevnik {
    display: grid;
    grid-template-columns: repeat(auto-fill, minmax(300px, 1fr));
    gap: 30px;
    margin-top: 40px;
}

.zapis {
    background: white;
    border-radius: 15px;
    overflow: hidden;
    box-shadow: 0 8px 20px rgba(0,0,0,0.1);
    transition: transform 0.3s;
}

.zapis:hover {
    transform: translateY(-5px);
    box-shadow: 0 15px 30px rgba(0,0,0,0.15);
}

.foto-container {
    height: 200px;
    overflow: hidden;
}

.zapis img {
    width: 100%;
    height: 100%;
    object-fit: cover;
    transition: transform 0.5s;
}

.zapis:hover img {
    transform: scale(1.05);
}

.podpis {
    padding: 20px;
}
```

```
.data {  
    display: inline-block;  
    background: #98fb98;  
    color: #006400;  
    padding: 5px 12px;  
    border-radius: 20px;  
    font-size: 14px;  
    font-weight: bold;  
    margin-bottom: 10px;  
}  
  
.nazvanie {  
    color: #8b4513;  
    font-size: 20px;  
    margin: 10px 0;  
    font-weight: bold;  
}  
  
.opisanie {  
    color: #556b2f;  
    line-height: 1.5;  
    font-size: 16px;  
}  
  
.mesto {  
    margin-top: 15px;  
    color: #666;  
    font-style: italic;  
    font-size: 14px;  
}  
  
.mesto:before {  
    content: "📍 ";  
}  
  
/* Особые стили для разных типов записей */  
.cvetok .data { background: #ffb6c1; color: #8b0000; }  
.urozhaj .data { background: #ffd700; color: #8b4513; }  
.peyzaj .data { background: #87ceeb; color: #00008b; }  
 </style>  
</head>  
<body>  
    <h1>🌿 Мой садовый дневник</h1>
```

```
<div class="dnevnik">
    <!-- Запись 1 -->
    <div class="zapis cvetok">
        <div class="foto-container">
            
        </div>
        <div class="podpis">
            <div class="data">15 мая 2024</div>
            <div class="nazvanie">Первые розы сезона</div>
            <p class="opisanie">Наконец-то расцвели розы сорта "Глория Дей". Аромат
стоит на весь сад! Пчёлы уже облюбовали их.</p>
            <div class="mesto">Центральная клумба</div>
        </div>
    </div>

    <!-- Запись 2 -->
    <div class="zapis urozhaj">
        <div class="foto-container">
            
        </div>
        <div class="podpis">
            <div class="data">10 июня 2024</div>
            <div class="nazvanie">Первый урожай клубники</div>
            <p class="opisanie">Собрали первую корзинку клубники. Ягоды крупные,
сладкие, просто тают во рту. Внуки съели всё за 5 минут!</p>
            <div class="mesto">Грядка у теплицы</div>
        </div>
    </div>

    <!-- Запись 3 -->
    <div class="zapis рейзаж">
        <div class="foto-container">
            
        </div>
        <div class="podpis">
            <div class="data">5 июля 2024</div>
            <div class="nazvanie">Утренняя роса в саду</div>
            <p class="opisanie">Встала на рассвете, чтобы запечатлеть эту красоту. Каждая
паутинка в росе переливается, как бриллиант.</p>
            <div class="mesto">Весь сад</div>
        </div>
    </div>
```

```
</div>
</div>

<div style="margin-top: 40px; padding: 20px; background: #ffffaf0; border-radius: 10px; border-left: 5px solid #daa520;">
    <h3> Что мы использовали для подписей:</h3>
    <ul>
        <li><strong>Структура:</strong> Каждая запись — отдельный блок с фото и текстовой частью</li>
        <li><strong>Дата:</strong> Выделена цветным фоном и скруглёнными углами</li>
        <li><strong>Название:</strong> Крупный жирный шрифт, контрастный цвет</li>
        <li><strong>Описание:</strong> Основной текст, удобный для чтения</li>
        <li><strong>Место:</strong> Мелкий курсив с иконкой-эмодзи</li>
        <li><strong>Цветовые акценты:</strong> Разные цвета для разных типов записей</li>
    </ul>
</div>
</body>
</html>
```

---

## 6. Важные правила хороших подписей

1. **Краткость:** Подпись должна быть информативной, но не слишком длинной.

Плохо: "Это фотография моей розы, которая растёт в саду уже три года, я посадила её весной 2021 года, она очень красивая и пахнет чудесно..."

Хорошо: "Роза "Гlorия Дей". Третий год цветения."

2. **Структура:** Разделяйте информацию:

text

Название/Заголовок

Основное описание

Дата и место

3. **Читаемость:**

- Достаточный контраст с фоном
- Умеренный размер шрифта (не слишком мелкий)
- Правильное выравнивание

4. **Доступность:** Всегда заполняйте alt у изображений:

html

```

```

5. **Единство стиля:** Если на сайте несколько фотографий, делайте подписи в одном стиле.
- 

## 7. Практическое задание

### Задание 1: "Семейный альбом"

1. Создайте страницу с 3-4 семейными фотографиями
2. Для каждой добавьте подпись с:
  - Дата события
  - Кто на фото
  - Краткое описание

### Задание 2: "Рецепт с фото"

2. Создайте страницу рецепта

3. Добавьте фото готового блюда

4. Под фото разместите:

- Название блюда
- Время приготовления
- Советы от автора

### Задание 3: "Интерактивная галерея"

1. Создайте 3 фотографии с подписями, которые появляются при наведении

2. Сделайте так, чтобы при наведении:

- Фото слегка увеличивалось
- Появлялась подпись
- Добавлялась тень

### Бонус: Подпись с цитатой:

html

```
<div class="foto-s-citatoy">
    
    <div class="citata">
        <p>"Осень — вторая весна, когда каждый лист — цветок."</p>
        <p class="avtor">— Альбер Камю</p>
    </div>
```

```
</div>  
CSS  
.foto-s-citatoj {  
    position: relative;  
}  
  
.citata {  
    position: absolute;  
    top: 50%;  
    left: 50%;  
    transform: translate(-50%, -50%);  
    background: rgba(255, 255, 255, 0.9);  
    padding: 20px;  
    border-radius: 10px;  
    text-align: center;  
    font-style: italic;  
    max-width: 80%;  
    box-shadow: 0 5px 15px rgba(0,0,0,0.2);  
}
```

---

## 8. Частые ошибки и как их избежать

### 1. Текст поверх важных деталей фото

-  Подпись закрывает лица на фото  
 Размещайте подпись в однотонной области или с полупрозрачным фоном

### 2. Слишком мелкий шрифт

-  font-size: 10px;  
 font-size: 14px-16px минимум

### 3. Отсутствие отступов

-  Текст прилипает к краям  
 Добавляйте padding: 10px-20px;

### 4. Неинформативные подписи

-  "Фото 1", "IMG\_1234.jpg"  
 Описывайте, что на фото и почему это важно
- 

**Дорогой друг, теперь ваши фотографии будут не просто картинками, а настоящими историями с контекстом!** Подпись — это как тихий голос,

который рассказывает зрителю: "Посмотри сюда, это важно, это красиво, это память".

**В следующей главе** мы освоим CSS Flexbox — волшебный инструмент, который поможет вам расставлять фотографии, тексты и другие элементы на странице так же легко, как расставляете мебель в комнате!

## ● Глава 11: Секрет гибких страниц – CSS Flexbox (объясняем без страшных слов!).

- Простая аналогия: Контейнер (**ящик для инструментов**) и гибкие элементы (**инструменты внутри**).

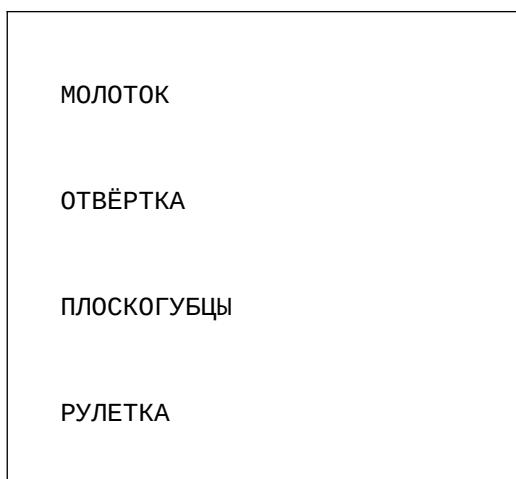
Дорогой друг, представьте, что у вас есть **ящик для инструментов** (контейнер), а внутри него лежат **инструменты** (гибкие элементы). Flexbox — это как умный ящик, который сам решает, как лучше разложить инструменты!

---

### ⌚ Аналогия «Ящик с инструментами»

Давайте представим это наглядно:

#### 1. Обычный ящик (без Flexbox):



Все инструменты свалены в кучу, каждый занимает всю ширину ящика. Чтобы взять отвёртку, нужно разгрести всё сверху.

#### 2. Умный ящик (с Flexbox):

МОЛОТОК	ОТВЁРТКА	ПЛОСКОГУБЦЫ
РУЛЕТКА	ГАЕЧНЫЙ КЛЮЧ	НОЖНИЦЫ

Инструменты аккуратно разложены в ряд! Ящик сам решил, как их лучше разместить.

---

## Перевод на язык CSS:

В ящике с инструментами	В CSS Flexbox	Код CSS
Ящик (контейнер)	Flex-контейнер	display: flex;
Инструменты внутри	Flex-элементы	Все дети контейнера
Раскладываем в ряд	Горизонтальное размещение	flex-direction: row;
Раскладываем в столбик	Вертикальное размещение	flex-direction: column;
Выравниваем по центру ящика	Выравнивание по центру	justify-content: center;
Равномерно распределяем	Равные промежутки	justify-content: space-between;

## Практический пример: Организуем инструменты в ящике

Давайте создадим наш «ящик для инструментов» на сайте:

```
html
<!DOCTYPE html>
<html lang="ru">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Мой виртуальный ящик с инструментами</title>
  <style>
```

```
/* ЯЩИК (контейнер) */
.yashik-dlya-instrumentov {
    display: flex; /* Вот она, волшебная строка! */
    flex-direction: row; /* Раскладываем в ряд */
    justify-content: space-around; /* Равномерно распределяем */
    align-items: center; /* Выравниваем по центру вертикально */

    /* Внешний вид ящика */
    width: 800px;
    height: 200px;
    background-color: #8b4513; /* Цвет дерева */
    border-radius: 15px;
    padding: 20px;
    margin: 50px auto;
    border: 5px solid #654321;
    box-shadow: 0 10px 30px rgba(0,0,0,0.3);
}

/* ИНСТРУМЕНТЫ (элементы) */
.instrument {
    width: 100px;
    height: 100px;
    background-color: #d2691e; /* Металлический цвет */
    border-radius: 10px;
    display: flex;
    justify-content: center;
    align-items: center;
    color: white;
    font-weight: bold;
    font-size: 18px;
    box-shadow: inset 0 0 10px rgba(0,0,0,0.5);
    transition: all 0.3s;
}

/* Эффект при наведении на инструмент */
.instrument:hover {
    transform: translateY(-10px);
    background-color: #a0522d;
    box-shadow: 0 10px 20px rgba(0,0,0,0.4);
}

/* Иконки инструментов (просто символы) */
.molotok::before { content: "🔨"; font-size: 40px; }
.otvyortka::before { content: "🔧"; font-size: 40px; }
```

```
.pilka::before { content: " "; font-size: 40px; }
.klyuch::before { content: " "; font-size: 40px; }
.rulon::before { content: " "; font-size: 40px; }

/* Подписи */
.podpis {
    margin-top: 5px;
    font-size: 14px;
    text-align: center;
}

/* Объясняющая панель */
.obyasnenie {
    background: #f9f5f0;
    padding: 25px;
    border-radius: 15px;
    max-width: 800px;
    margin: 30px auto;
    border-left: 6px solid #8b4513;
}

.obyasnenie h3 {
    color: #8b4513;
    margin-top: 0;
}

.klyuchevoe-slovo {
    background: #ffd700;
    padding: 2px 6px;
    border-radius: 4px;
    font-weight: bold;
}

</style>
</head>
<body>
<h1 style="text-align: center; color: #8b4513;">🔧 Мой волшебный ящик с инструментами</h1>

<!-- ЯЩИК (FLEX-КОНТЕЙНЕР) -->
<div class="yashik-dlya-instrumentov">
    <!-- ИНСТРУМЕНТЫ (FLEX-ЭЛЕМЕНТЫ) -->
    <div>
        <div class="instrument molotok"></div>
        <div class="podpis">Молоток</div>
    </div>

```

```

<div>
  <div class="instrument otvyortka"></div>
  <div class="podpis">Отвёртка</div>
</div>

<div>
  <div class="instrument pilka"></div>
  <div class="podpis">Пилка</div>
</div>

<div>
  <div class="instrument klyuch"></div>
  <div class="podpis">Гаечный ключ</div>
</div>

<div>
  <div class="instrument rulon"></div>
  <div class="podpis">Рулетка</div>
</div>
</div>

<div class="obyasnenie">
  <h3>⌚ Как это работает (простыми словами):</h3>

  <p><span class="klyuchchevoe-slovo">1. Ящик (контейнер):</span> Див с классом
<code>.yashik-dlya-instrumentov</code></p>
  <p><span class="klyuchchevoe-slovo">2. Волшебное заклинание:</span> <code>display:
flex;</code></p>
  <p>Этой одной строчкой мы сказали ящику: <em>"Теперь ты умный! Раскладывай всё,
что внутри, аккуратно!"</em></p>

  <h3>📦 Что умеет наш ящик:</h3>
  <ul>
    <li><strong>Раскладывать в ряд:</strong> <code>flex-direction: row;</code>
(горизонтально)</li>
    <li><strong>Раскладывать в столбик:</strong> <code>flex-direction: column;</code>
(вертикально)</li>
    <li><strong>Распределять равномерно:</strong> <code>justify-content: space-
around;</code></li>
    <li><strong>Выравнивать по центру:</strong> <code>align-items: center;</code></li>
  </ul>

  <h3>🔄 Попробуйте изменить:</h3>

```

```

<p>В CSS-коде поменяйте:</p>
<ul>
    <li><code>flex-direction: row;</code> на <code>flex-direction: column;</code> —
        инструменты встанут в столбик!</li>
    <li><code>justify-content: space-around;</code> на <code>justify-content:
        center;</code> — сберутся в центре!</li>
    <li>Уберите <code>display: flex;</code> — и всё "развалится"!</li>
</ul>
</div>

<!-- Дополнительный пример: разные раскладки --&gt;
&lt;div style="display: flex; gap: 20px; max-width: 800px; margin: 50px auto;"&gt;
    &lt;div style="flex: 1; background: #e8f4fc; padding: 20px; border-radius: 10px;"&gt;
        &lt;h4&gt;🔧 Ряд (row)&lt;/h4&gt;
        &lt;div style="display: flex; background: #d2b48c; padding: 10px; border-radius: 5px;"&gt;
            &lt;div style="width: 30px; height: 30px; background: #8b4513; margin: 5px;"&gt;&lt;/div&gt;
            &lt;div style="width: 30px; height: 30px; background: #8b4513; margin: 5px;"&gt;&lt;/div&gt;
            &lt;div style="width: 30px; height: 30px; background: #8b4513; margin: 5px;"&gt;&lt;/div&gt;
        &lt;/div&gt;
        &lt;p&gt;Инструменты в ряд, как на полке.&lt;/p&gt;
    &lt;/div&gt;

    &lt;div style="flex: 1; background: #f0fff0; padding: 20px; border-radius: 10px;"&gt;
        &lt;h4&gt;📋 Столбик (column)&lt;/h4&gt;
        &lt;div style="display: flex; flex-direction: column; background: #98fb98; padding: 10px;
            border-radius: 5px; height: 150px;"&gt;
            &lt;div style="width: 30px; height: 30px; background: #2e8b57; margin: 5px;"&gt;&lt;/div&gt;
            &lt;div style="width: 30px; height: 30px; background: #2e8b57; margin: 5px;"&gt;&lt;/div&gt;
            &lt;div style="width: 30px; height: 30px; background: #2e8b57; margin: 5px;"&gt;&lt;/div&gt;
        &lt;/div&gt;
        &lt;p&gt;Инструменты в столбик, как в коробке.&lt;/p&gt;
    &lt;/div&gt;
&lt;/div&gt;
&lt;/body&gt;
&lt;/html&gt;
</pre>


---



```



## Ключевые понятия Flexbox в аналогии с ящиком:

### 1. `display: flex;` — «Включить умный режим»

css

```
.yashik {  
    display: flex; /* Теперь ящик умный! */  
}
```

Без этой строчки инструменты просто свалены в кучу. С ней — ящик сам их организует.

## 2. **flex-direction** — «Как раскладывать»

css

```
.yashik {  
    flex-direction: row; /* В ряд (слева направо) */  
    /* или */  
    flex-direction: column; /* В столбик (сверху вниз) */  
}
```

## 3. **justify-content** — «Распределить вдоль ящика»

css

```
.yashik {  
    justify-content: flex-start; /* К левому краю */  
    justify-content: center; /* В центр */  
    justify-content: flex-end; /* К правому краю */  
    justify-content: space-between; /* По краям с промежутками */  
    justify-content: space-around; /* С отступами со всех сторон */  
}
```

## 4. **align-items** — «Выровнять попрёк ящика»

css

```
.yashik {  
    align-items: stretch; /* Растянуть (по умолчанию) */  
    align-items: flex-start; /* К верхнему краю */  
    align-items: center; /* По центру */  
    align-items: flex-end; /* К нижнему краю */  
}
```

---

## Реальный пример: Расставляем мебель в комнате

Давайте применим аналогию к чему-то ещё более понятному — расстановке мебели!

html

```
<div class="komnata">
    <div class="divan">Диван</div>
    <div class="stol">Стол</div>
    <div class="stul">Стул</div>
    <div class="shkaf">Шкаф</div>
</div>
```

css

```
.komnata {
    display: flex; /* Комната становится умной! */
    flex-direction: row; /* Мебель в ряд вдоль стены */
    justify-content: space-between; /* Равномерно по стене */
    align-items: center; /* Вся мебель на одном уровне */

    width: 600px;
    height: 150px;
    background-color: #f5f5dc; /* Цвет стен */
    border: 3px solid #8b4513;
    padding: 20px;
}
```

---

## ⌚ Практическое задание: «Создай свой ящик!»

### Задание 1: Ящик для рукоделия

1. Создайте контейнер с `display: flex;`
2. Добавьте 4 элемента: "Нитки", "Иголки", "Пуговицы", "Ленты"
3. Расположите их в ряд с равными промежутками

### Задание 2: Книжная полка

1. Создайте вертикальный контейнер (`flex-direction: column`)
2. Добавьте книги: "Роман", "Детектив", "Поэзия", "Справочник"
3. Выровняйте по центру

### **Задание 3: Фото-галерея**

1. Создайте flex-контейнер
2. Добавьте 5 фотографий (можно цветные блоки)
3. Сделайте так, чтобы фотографии переносились на новую строку, если не помещаются

### **Бонус: Интерактивный ящик**

html

```
<button onclick="document.getElementById('yashik').style.flexDirection = 'column'">  
    Поставить в столбик  
</button>  
<button onclick="document.getElementById('yashik').style.justifyContent = 'center'">  
    Собрать в центре  
</button>
```

---



### **Главные выводы:**

1. **Flexbox = умный контейнер**, который сам управляет своими детьми
2. **Всего одна строчка** `display: flex;` творит чудеса
3. **Контейнер командует**, элементы подчиняются
4. **Аналогия с ящиком** помогает запомнить:
  - Контейнер = ящик
  - Элементы = инструменты
  - `flex-direction` = как раскладывать (в ряд/столбик)
  - `justify-content` = как распределять вдоль
  - `align-items` = как выравнивать поперёк

**Дорогой друг, теперь у вас есть волшебный ящик (Flexbox), который может аккуратно раскладывать всё, что вы в него положите!** Это как иметь личного помощника по вёрстке.

**В следующей части главы** мы создадим реальную галерею фотографий с помощью Flexbox — вы увидите, как несколько строчек кода создают красивый, адаптивный макет!

- Всего одно свойство `display: flex` для контейнера – и элементы выстраиваются в ряд или столбик, становятся гибкими!

Дорогой друг, сейчас вы узнаете о **самом волшебном свойстве CSS!**

Представьте, что у вас есть волшебная палочка. Вы говорите: «Раз, два, три — станьте гибкими!» — и все элементы на странице начинают слушаться вас. Эта волшебная палочка и есть `display: flex`!

---

## ⌚ Магия одной строчки: `display: flex`

ДО применения `display: flex`:

html

```
<div class="контейнер">
    <div class="элемент">Блок 1</div>
    <div class="элемент">Блок 2</div>
    <div class="элемент">Блок 3</div>
</div>
```

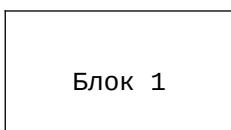
css

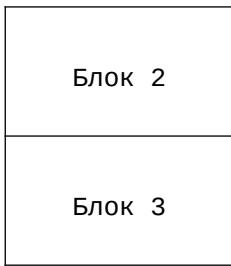
```
.контейнер {
    /* Пусто! Никакой магии */
    border: 3px solid #8b4513;
    padding: 20px;
}
```

```
.элемент {
    background: #3498db;
    color: white;
    padding: 20px;
    margin: 5px;
}
```

Что получится:

text





Каждый блок занимает всю ширину и сваливается друг на друга, как книги в старой сумке.

---

### ПОСЛЕ применения `display: flex;`:

Просто добавляем **ОДНУ СТРОЧКУ**:

css

```
.контейнер {  
    display: flex; /* ★ ВОТ ОНА, МАГИЯ! ★ */  
    border: 3px solid #8b4513;  
    padding: 20px;  
}
```

**Что получится:**

text



Все три блока **мгновенно** выстроились в ряд! Они стали **гибкими** — подстроились под размер контейнера.

---

### 🔍 Почему это волшебство? Что происходит на самом деле?

Когда вы пишете `display: flex;`, вы говорите браузеру:

«Дорогой браузер! Всем моим детям (элементам внутри этого контейнера) отныне запрещаю вести себя как обычные блоки. Пусть они станут **гибкими детьми** и слушаются новых правил!»

## Автоматически включаются «умные» настройки:

1. Элементы выстраиваются в ряд (по умолчанию flex-direction: row)
  2. Элементы не переносятся (по умолчанию flex-wrap: nowrap)
  3. Элементы начинаются с левого края (по умолчанию justify-content: flex-start)
  4. Элементы растягиваются по высоте (по умолчанию align-items: stretch)
- 



## Полный пример с визуализацией «До» и «После»

html

```
<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Магия display: flex</title>
    <style>
        /* Общие стили */
        body {
            font-family: 'Georgia', serif;
            background: #f9f5f0;
            padding: 30px;
            max-width: 1000px;
            margin: 0 auto;
        }

        h1 {
            text-align: center;
            color: #8b4513;
            margin-bottom: 40px;
        }

        /* Сравнение ДО и ПОСЛЕ */
        .sravnenie {
            display: flex;
            gap: 40px;
            margin-bottom: 50px;
        }
    </style>
</head>
<body>
    <h1>Hello world!</h1>
    <div class="sravnenie">
        <div>Блок 1</div>
        <div>Блок 2</div>
        <div>Блок 3</div>
    </div>
</body>
</html>
```

```
}

.panel {
    flex: 1;
    background: white;
    padding: 25px;
    border-radius: 15px;
    box-shadow: 0 10px 25px rgba(0,0,0,0.1);
}

.panel h2 {
    color: #2c3e50;
    text-align: center;
    margin-top: 0;
    padding-bottom: 15px;
    border-bottom: 3px solid;
}

/* Кон테йнеры для демонстрации */

.konteyner {
    border: 3px dashed #8b4513;
    padding: 20px;
    margin: 20px 0;
    min-height: 150px;
    background: #ffffaf5;
}

.bez-flex { border-color: #e74c3c; }
.bez-flex h2 { border-color: #e74c3c; }

.s-flex { border-color: #2ecc71; }
.s-flex h2 { border-color: #2ecc71; }

/* Элементы внутри контейнеров */

.element {
    background: #3498db;
    color: white;
    padding: 25px;
    font-weight: bold;
    font-size: 20px;
    text-align: center;
    border-radius: 8px;
    margin: 5px;
    transition: all 0.3s;
```

```
}

/* Цвета для разных элементов */
.el1 { background: #e74c3c; } /* Красный */
.el2 { background: #2ecc71; } /* Зелёный */
.el3 { background: #3498db; } /* Синий */
.el4 { background: #f39c12; } /* Оранжевый */

/* ★ ВОЛШЕБНАЯ СТРОЧКА ★ */
.magicheskiy-konteyner {
    display: flex; /* ВСЁ МЕНЯЕТСЯ ЗДЕСЬ! */
}

/* Панель управления */
.upravlenie {
    background: #2c3e50;
    color: white;
    padding: 30px;
    border-radius: 15px;
    margin: 30px 0;
}

.knopka {
    background: #3498db;
    color: white;
    border: none;
    padding: 15px 25px;
    margin: 10px;
    border-radius: 8px;
    font-size: 16px;
    cursor: pointer;
    transition: all 0.3s;
}

.knopka:hover {
    background: #2980b9;
    transform: translateY(-3px);
}

.vazhno {
    background: #fff8e1;
    padding: 20px;
    border-radius: 10px;
    margin: 20px 0;
```

```
border-left: 6px solid #ffb300;
}

.volshebnaya-stroka {
background: #2c3e50;
color: #ffd700;
padding: 15px;
border-radius: 10px;
font-family: 'Courier New', monospace;
font-size: 24px;
text-align: center;
margin: 20px 0;
animation: pulse 2s infinite;
}

@keyframes pulse {
0% { box-shadow: 0 0 10px #ffd700; }
50% { box-shadow: 0 0 25px #ffd700; }
100% { box-shadow: 0 0 10px #ffd700; }
}

.kod {
background: #f8f9fa;
padding: 15px;
border-radius: 8px;
font-family: 'Courier New', monospace;
border-left: 4px solid #3498db;
margin: 15px 0;
}

</style>
</head>
<body>
<h1>☀️ Магия одной строчки: display: flex</h1>

<div class="volshebnaya-stroka">
display: flex;
</div>

<div class="vazhno">
<p><strong>💡 Важно понять:</strong> Вся магия происходит с контейнером (родителем), а не с элементами (детьми)!</p>
<p>Вы говорите родителю: «Стань flex-контейнером!» — а его дети автоматически становятся flex-элементами.</p>
</div>
```

```
<div class="sravnenie">
    <!-- БЕЗ FLEX -->
    <div class="panel bez-flex">
        <h2>✗ БЕЗ display: flex</h2>
        <p>Элементы ведут себя как обычные блоки:</p>

        <div class="konteyner">
            <div class="element el1">1</div>
            <div class="element el2">2</div>
            <div class="element el3">3</div>
            <div class="element el4">4</div>
        </div>

        <div class="kod">
            .контейнер {<br>
                &nbsp;&nbsp;*> Ничего особенного! *<br>
            }
        </div>

        <p><strong>Что происходит:</strong></p>
        <ul>
            <li>Каждый блок занимает всю ширину</li>
            <li>Блоки сваливаются друг на друга</li>
            <li>Нет контроля над расположением</li>
            <li>Жёсткая, негибкая структура</li>
        </ul>
    </div>

    <!-- С FLEX -->
    <div class="panel s-flex">
        <h2>✓ С display: flex</h2>
        <p>Всего одна волшебная строчка:</p>

        <div class="konteyner magicheskiy-konteyner">
            <div class="element el1">1</div>
            <div class="element el2">2</div>
            <div class="element el3">3</div>
            <div class="element el4">4</div>
        </div>

        <div class="kod">
            .контейнер {<br>
                &nbsp;&nbsp;<strong>display: flex;</strong> ← ★ Магия здесь! ★<br>

```

```

        }
    </div>

    <p><strong>Что изменилось:</strong></p>
    <ul>
        <li>Блоки выстроились в ряд автоматически</li>
        <li>Блоки стали гибкими (скимаются/растягиваются)</li>
        <li>Появился контроль над расположением</li>
        <li>Структура стала адаптивной</li>
    </ul>
</div>
</div>


<div class="upravlenie">
    <h2 style="color: white; text-align: center;">🎮 Управляй магией!</h2>

    <div style="text-align: center;">
        <button class="knopka" onclick="primenitFlex()">✨ Применить display: flex</button>
        <button class="knopka" onclick="ubratFlex()">✖️ Убрать display: flex</button>
        <button class="knopka" onclick="pomenyatNapravlenie()">🔄 Изменить
            направление</button>
        <button class="knopka" onclick="dobavitElement()">➕ Добавить элемент</button>
    </div>

    <div id="demo-konteyner" style="margin-top: 30px; padding: 20px; background:
        rgba(255,255,255,0.1); border-radius: 10px;">
        <div id="demo-elementy" style="display: flex; gap: 10px; min-height: 100px; align-items:
            center; justify-content: center;">
            <div class="element el1" style="padding: 20px;">A</div>
            <div class="element el2" style="padding: 20px;">B</div>
            <div class="element el3" style="padding: 20px;">C</div>
        </div>

        <div style="text-align: center; margin-top: 20px; color: #ddd;">
            <p id="status">Текущее состояние: <strong>display: flex</strong> (элементы в
                ряд)</p>
            <p><small>Нажмите кнопки и смотрите, как меняется расположение
                элементов!</small></p>
        </div>
    </div>
</div>

<!-- Что ещё можно делать с display: flex --&gt;</pre>

```

```

<div style="background: white; padding: 30px; border-radius: 15px; margin: 30px 0;">
    <h2 style="color: #2c3e50;">💡 Что ещё умеет display: flex?</h2>

    <div style="display: flex; gap: 20px; margin-top: 30px;">
        <div style="flex: 1; padding: 20px; background: #e8f4fc; border-radius: 10px;">
            <h3>📦 Гибкие размеры</h3>
            <div style="display: flex; background: #d2e3f7; padding: 15px; border-radius: 8px;
height: 100px;">
                <div style="flex: 1; background: #3498db; margin: 5px;"></div>
                <div style="flex: 2; background: #2980b9; margin: 5px;"></div>
                <div style="flex: 1; background: #3498db; margin: 5px;"></div>
            </div>
            <p>Элементы могут иметь разную «гибкость» (flex-grow)</p>
        </div>

        <div style="flex: 1; padding: 20px; background: #f0fff0; border-radius: 10px;">
            <h3>🔄 Перенос строк</h3>
            <div style="display: flex; flex-wrap: wrap; background: #c8e6c9; padding: 15px;
border-radius: 8px;">
                <div style="width: 60px; height: 60px; background: #2ecc71; margin:
5px;"></div>
                <div style="width: 60px; height: 60px; background: #2ecc71; margin:
5px;"></div>
                <div style="width: 60px; height: 60px; background: #2ecc71; margin:
5px;"></div>
                <div style="width: 60px; height: 60px; background: #2ecc71; margin:
5px;"></div>
            </div>
            <p>Элементы переносятся на новую строку (flex-wrap)</p>
        </div>
    </div>
</div>

<!-- Практическое применение -->
<div style="background: linear-gradient(135deg, #667eea 0%, #764ba2 100%); color: white;
padding: 40px; border-radius: 15px; margin: 30px 0;">
    <h2 style="color: white; text-align: center;">🏡 Практическое применение</h2>

    <p style="text-align: center; font-size: 18px;">Где использовать display: flex в реальных
проектах:</p>

    <div style="display: flex; gap: 20px; margin-top: 30px; text-align: center;">
        <div style="flex: 1; background: rgba(255,255,255,0.1); padding: 20px; border-radius:
10px; backdrop-filter: blur(10px);">

```

```
<div style="font-size: 40px;">  </div>
<h3>Навигационное меню</h3>
<p>Кнопки меню автоматически выстраиваются в ряд</p>
</div>

<div style="flex: 1; background: rgba(255,255,255,0.1); padding: 20px; border-radius: 10px; backdrop-filter: blur(10px);">
    <div style="font-size: 40px;">  </div>
    <h3>Фотогалерея</h3>
    <p>Фотографии аккуратно располагаются в сетке</p>
</div>

<div style="flex: 1; background: rgba(255,255,255,0.1); padding: 20px; border-radius: 10px; backdrop-filter: blur(10px);">
    <div style="font-size: 40px;">  </div>
    <h3>Карточки товаров</h3>
    <p>Карточки одинаковой высоты, даже если текст разный</p>
</div>
</div>

<script>
let demoKonteyner = document.getElementById('demo-elementy');
let statusText = document.getElementById('status');
let napravlenie = 'row';
let imetFlex = true;

function primenitFlex() {
    demoKonteyner.style.display = 'flex';
    imetFlex = true;
    obnovitStatus();
}

function ubratFlex() {
    demoKonteyner.style.display = 'block';
    imetFlex = false;
    obnovitStatus();
}

function pomenyatNapravlenie() {
    if (napravlenie === 'row') {
        demoKonteyner.style.flexDirection = 'column';
        napravlenie = 'column';
    } else {
```

```
        demoKonteyner.style.flexDirection = 'row';
        napravlenie = 'row';
    }
    obnovitStatus();
}

function dobavitElement() {
    const noviyElement = document.createElement('div');
    const bukvy = ['D', 'E', 'F', 'G', 'H'];
    const cveta = ['#9b59b6', '#e67e22', '#1abc9c', '#e74c3c', '#f1c40f'];

    const bukva = bukvy[Math.floor(Math.random() * bukvy.length)];
    const cvet = cveta[Math.floor(Math.random() * cveta.length)];

    noviyElement.className = 'element';
    noviyElement.textContent = bukva;
    noviyElement.style.background = cvet;
    noviyElement.style.padding = '20px';
    noviyElement.style.borderRadius = '8px';

    demoKonteyner.appendChild(noviyElement);
    obnovitStatus();
}

function obnovitStatus() {
    let text = 'Текущее состояние: ';

    if (!imeetFlex) {
        text += '<strong>display: block</strong> (обычные блоки)';
    } else if (napravlenie === 'row') {
        text += '<strong>display: flex</strong> (элементы в ряд)';
    } else {
        text += '<strong>display: flex</strong> (элементы в столбик)';
    }

    text += ` | Элементов: ${demoKonteyner.children.length}`;
    statusText.innerHTML = text;
}

// Инициализация
obnovitStatus();

</script>
</body>
</html>
```

---

## 🎯 Ключевые моменты, которые нужно запомнить:

### 1. Магия работает СВЕРХУ ВНИЗ

css

```
/* Правильно: применяем к родителю */
.roditel {
    display: flex; /* Дети станут flex-элементами */
}

.deti {
    /* Ничего не нужно писать! */
    /* Они УЖЕ стали flex-элементами */
}
```

### 2. Автоматические изменения после `display: flex`:

- Элементы становятся в ряд
- Элементы перестают быть блочными
- Исчезают лишние отступы между inline-элементами
- Появляется контроль над выравниванием

### 3. На что влияет `display: flex`:

#### Было (без flex)

div занимает всю ширину

Элементы сваливаются друг на друга

Нет контроля над вертикальным  
выравниванием

Сложно делать адаптивные колонки

#### Стало (с flex)

div занимает только нужное место

Элементы выстраиваются в ряд

Легко выравнивать по центру  
вертикально

Колонки создаются одной строчкой

## Быстрые примеры использования одной строчки `display: flex`:

### Пример 1: Центрирование по горизонтали и вертикали (раньше это было сложно!)

css

```
.centrirovat {  
    display: flex;      /* ★ Волшебство! ★ */  
    justify-content: center; /* По горизонтали */  
    align-items: center; /* По вертикали */  
    height: 300px;  
}
```

### Пример 2: Навигационное меню за 3 секунды

css

```
.menu {  
    display: flex; /* Готово! Кнопки в ряд */  
}
```

*/\* Без flex пришлось бы писать: \*/*

```
.menu a {  
    display: inline-block;  
    float: left;  
    margin-right: 10px;  
    /* И ёщё 10 строк кода... */  
}
```

### Пример 3: Карточки одинаковой высоты

css

```
.kartochki {  
    display: flex; /* Все карточки станут одной высоты! */  
}
```

---

## Практическое задание:

### Задание 1: Создайте меню за 1 минуту

1. Создайте div с классом menu
2. Добавьте внутрь 5 ссылок: "Главная", "О нас", "Услуги", "Фото", "Контакты"
3. Добавьте в CSS: .menu { display: flex; }
4. Посмотрите результат!

## Задание 2: Центрируйте заголовок

1. Создайте блок высотой 200px
2. Добавьте внутри заголовок h1
3. Добавьте CSS:

css

```
.blok {  
    display: flex;  
    justify-content: center;  
    align-items: center;  
}
```

## Задание 3: Сравните «До» и «После»

1. Создайте два одинаковых блока с 4 элементами
2. К первому блоку не применяйте flex
3. Ко второму блоку примените display: flex
4. Убедитесь в разнице!

## Бонус: Интерактивный переключатель

javascript

```
//Добавьте эту кнопку на страницу  
<button onclick="document.querySelector('.blok').style.display = 'flex'">  
    Включить магию Flex!  
</button>
```

---

## ❖ Золотые правила display: flex:

1. Одна строчка решает 90% проблем с расположением элементов
2. Применяется только к родителю — дети подчиняются автоматически
3. Не ломает ничего — всегда можно убрать, и всё вернётся как было
4. Работает во всех современных браузерах (даже в старых версиях)
5. Идеально для: меню, галерей, карточек, форм, футеров, хедеров

**Дорогой друг, теперь у вас в руках настоящая волшебная палочка веб-дизайна!** Одна строчка `display: flex` превращает хаотично расположенные элементы в аккуратную, управляемую композицию.

**Запомните:** Когда видите, что элементы нужно расположить в ряд, в столбик, выровнять или распределить — первое, что нужно попробовать, это `display: flex`!

**В следующей части** мы сделаем реальную фотогалерею с помощью Flexbox — вы увидите, как несколько свойств создают профессиональный макет!

## ■ Практикум №7: Создаем простую галерею фотографий, которая аккуратно выстраивается сама.

Дорогой друг, представьте, что у вас есть коллекция фотографий внуков, отпуска или вашего сада. Вы хотите показать их на сайте, но не просто свалкой, а красиво — чтобы они сами выстроились в аккуратные ряды, как солдаты на параде. С помощью Flexbox мы сделаем это за 5 минут!

---

### Шаг 1: Подготовка «цифрового фотоальбома»

#### 1.1. Создаем структуру проекта

Создайте папку Галерея\_Фотографий и в ней два файла:

1. **Файл** index.html — структура страницы
  2. **Файл** style.css — стили для галереи
- 

#### 1.2. Базовый HTML-код

```
html
<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Моя первая фото-галерея</title>
    <link rel="stylesheet" href="style.css">
    <link rel="stylesheet" href="https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css">
</head>
<body>
    <div class="container">
        <header class="header">
            <h1><i class="fas fa-camera"></i> Моя семейная галерея</h1>
```

```
<p class="subtitle">Фотографии самых дорогих моментов</p>
</header>

<main class="main-content">
    <!-- Галерея будет здесь -->
    <div class="gallery" id="gallery">
        <!-- Фотографии добавляются автоматически -->
    </div>

    <!-- Панель управления -->
    <div class="control-panel">
        <h2><i class="fas fa-sliders-h"></i> Управление галереей</h2>
        <div class="controls">
            <button class="btn" onclick="changeLayout('row')">
                <i class="fas fa-bars"></i> В ряд
            </button>
            <button class="btn" onclick="changeLayout('column')">
                <i class="fas fa-stream"></i> В столбик
            </button>
            <button class="btn" onclick="changeLayout('wrap')">
                <i class="fas fa-th"></i> Сетка
            </button>
            <button class="btn" onclick="addPhoto()">
                <i class="fas fa-plus"></i> Добавить фото
            </button>
            <button class="btn" onclick="changeTheme()">
                <i class="fas fa-palette"></i> Сменить тему
            </button>
        </div>
    </div>

    <!-- Объяснение как работает -->
    <div class="explanation">
        <h2><i class="fas fa-lightbulb"></i> Как это работает?</h2>
        <div class="code-example">
            <div class="code-block">
                <h3>HTML структура:</h3>
                <pre><code>&lt;div class="gallery"&gt;
&lt;div class="photo"&gt;
    &lt;img src="photo1.jpg"&gt;
    &lt;div class="caption"&gt;Подпись&lt;/div&gt;
&lt;/div&gt;
&lt;!-- еще фото --&gt;
&lt;/div&gt;</code></pre>

```

```

        </div>
        <div class="code-block">
            <h3>Волшебный CSS:</h3>
            <pre><code>.gallery {
display: flex;      /* Включаем Flexbox */
flex-wrap: wrap;    /* Разрешаем перенос */
justify-content: center; /* Центрируем */
gap: 20px;         /* Промежутки */
}

.photo {
    flex: 1 1 300px;    /* Гибкий размер */
    max-width: 400px;    /* Макс. ширина */
}</code></pre>
        </div>
    </div>
</div>
</main>

<footer class="footer">
    <p>Создано с ❤ вами с помощью CSS Flexbox</p>
    <p class="date">Сегодня: <span id="currentDate"></span></p>
</footer>
</div>

<script src="script.js"></script>
</body>
</html>

```

---

## Шаг 2: Создаем волшебные стили CSS

**Файл style.css:**

css

```

/* ===== ОБЩИЕ СТИЛИ ===== */
* {
    margin: 0;
    padding: 0;
    box-sizing: border-box;
}

```

```
body {
    font-family: 'Segoe UI', Tahoma, Geneva, Verdana, sans-serif;
    line-height: 1.6;
    color: #333;
    background: linear-gradient(135deg, #f5f7fa 0%, #c3cfe2 100%);
    min-height: 100vh;
    padding: 20px;
}

.container {
    max-width: 1200px;
    margin: 0 auto;
    background: white;
    border-radius: 20px;
    box-shadow: 0 15px 35px rgba(0, 0, 0, 0.1);
    overflow: hidden;
}

/* ===== ШАПКА ===== */
.header {
    background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);
    color: white;
    padding: 40px 30px;
    text-align: center;
    border-bottom: 5px solid #ffd700;
}

.header h1 {
    font-size: 2.8rem;
    margin-bottom: 10px;
    text-shadow: 2px 2px 4px rgba(0, 0, 0, 0.3);
}

.header .subtitle {
    font-size: 1.2rem;
    opacity: 0.9;
    font-style: italic;
}

.header i {
    margin-right: 15px;
    color: #ffd700;
}
```

```
/* ===== ГАЛЕРЕЯ - САМОЕ ВАЖНОЕ! ===== */

.gallery {
    padding: 40px 30px;

    /* ★★★ БОТ ОНА - ВОЛШЕБНАЯ СТРОЧКА! ★★★ */
    display: flex;      /* Включаем Flexbox */

    /* Гибкие настройки галереи */
    flex-wrap: wrap;    /* Разрешаем перенос на новую строку */
    justify-content: center; /* Центрируем фото по горизонтали */
    align-items: stretch; /* Все фото одной высоты */
    gap: 25px;          /* Промежутки между фото */

    /* Плавные переходы */
    transition: all 0.5s ease;
}

/* ===== ОТДЕЛЬНАЯ ФОТОГРАФИЯ ===== */

.photo {
    /* Гибкие размеры фото */
    flex: 1 1 300px;    /* flex-grow: 1, flex-shrink: 1, flex-basis: 300px */
    max-width: 400px;    /* Максимальная ширина */

    /* Оформление */
    background: white;
    border-radius: 15px;
    overflow: hidden;
    box-shadow: 0 10px 20px rgba(0, 0, 0, 0.15);
    transition: all 0.4s cubic-bezier(0.175, 0.885, 0.32, 1.275);
    border: 1px solid #eee;
}

/* Эффект при наведении */
.photo:hover {
    transform: translateY(-10px) scale(1.03);
    box-shadow: 0 20px 40px rgba(0, 0, 0, 0.2);
}

/* Само изображение */

.photo img {
    width: 100%;        /* Занимает всю ширину блока */
    height: 250px;       /* Фиксированная высота */
    object-fit: cover;   /* Обрезаем фото, чтобы оно красиво вписалось */
    display: block;       /* Убираем лишние отступы */
}
```

```
border-bottom: 3px solid #667eea;
}

/* Подпись к фото */
.caption {
    padding: 20px;
    background: #f8f9fa;
}

.caption h3 {
    color: #2c3e50;
    margin-bottom: 8px;
    font-size: 1.3rem;
}

.caption p {
    color: #666;
    font-size: 0.95rem;
    line-height: 1.5;
}

.caption .date {
    display: inline-block;
    background: #e8f4fc;
    color: #3498db;
    padding: 5px 12px;
    border-radius: 20px;
    font-size: 0.85rem;
    margin-top: 10px;
    font-weight: bold;
}

/* ===== ПАНЕЛЬ УПРАВЛЕНИЯ ===== */

.control-panel {
    background: #2c3e50;
    color: white;
    padding: 30px;
    margin: 0 30px 30px;
    border-radius: 15px;
    box-shadow: 0 10px 20px rgba(0, 0, 0, 0.2);
}

.control-panel h2 {
    margin-bottom: 25px;
```

```
    text-align: center;
    color: #ffd700;
}

.control-panel i {
    margin-right: 10px;
}

.controls {
    display: flex;
    flex-wrap: wrap;
    justify-content: center;
    gap: 15px;
}

.btn {
    display: flex;
    align-items: center;
    background: #3498db;
    color: white;
    border: none;
    padding: 15px 25px;
    border-radius: 50px;
    font-size: 1rem;
    font-weight: 600;
    cursor: pointer;
    transition: all 0.3s;
    box-shadow: 0 5px 15px rgba(52, 152, 219, 0.3);
}

.btn:hover {
    background: #2980b9;
    transform: translateY(-3px);
    box-shadow: 0 8px 20px rgba(52, 152, 219, 0.4);
}

.btn:active {
    transform: translateY(-1px);
}

/* ===== ОБЪЯСНЕНИЕ ===== */

.explanation {
    background: #f8f9fa;
    padding: 40px 30px;
```

```
margin: 0 30px 30px;
border-radius: 15px;
border-left: 6px solid #2ecc71;
}

.explanation h2 {
  color: #2c3e50;
  margin-bottom: 25px;
  text-align: center;
}

.code-example {
  display: flex;
  flex-wrap: wrap;
  gap: 30px;
  justify-content: center;
}

.code-block {
  flex: 1 1 400px;
  background: #2c3e50;
  color: #ecf0f1;
  padding: 25px;
  border-radius: 10px;
  font-family: 'Courier New', monospace;
  overflow-x: auto;
}

.code-block h3 {
  color: #2ecc71;
  margin-bottom: 15px;
  font-family: 'Segoe UI', sans-serif;
}

.code-block pre {
  white-space: pre-wrap;
  word-wrap: break-word;
}

.code-block code {
  color: #f8c555;
  font-size: 0.95rem;
  line-height: 1.5;
}
```

```
/* ===== ПОДВАЛ ===== */
.footer {
    background: #34495e;
    color: white;
    text-align: center;
    padding: 25px;
    border-top: 5px solid #ffd700;
}

.footer p {
    margin: 10px 0;
    font-size: 1rem;
}

.footer .date {
    color: #ffd700;
    font-weight: bold;
}

/* ===== РАЗНЫЕ РАСКЛАДКИ ГАЛЕРЕИ ===== */

/* Галерея в ряд (без переноса) */
.gallery-row {
    flex-wrap: nowrap !important; /* Запрещаем перенос */
    overflow-x: auto;           /* Добавляем прокрутку */
    padding-bottom: 20px;
    justify-content: flex-start;
}

/* Галерея в столбик */
.gallery-column {
    flex-direction: column !important;
    align-items: center;
    gap: 30px;
}

.gallery-column .photo {
    max-width: 600px !important;
    width: 100%;
}

/* Сетка (стандартный режим) */
.gallery-wrap {
```

```
flex-wrap: wrap;
justify-content: center;
}

/* ===== АДАПТИВНОСТЬ ===== */

@media (max-width: 768px) {
    .header h1 {
        font-size: 2rem;
    }

    .gallery {
        padding: 20px 15px;
        gap: 20px;
    }

    .photo {
        flex: 1 1 250px;
    }

    .controls {
        flex-direction: column;
    }

    .btn {
        justify-content: center;
    }
}

@media (max-width: 480px) {
    .header {
        padding: 25px 15px;
    }

    .photo {
        flex: 1 1 100%;
    }

    .code-example {
        flex-direction: column;
    }
}
```

---

## Шаг 3: Добавляем интерактивность JavaScript

**Файл** script.js:

```
javascript

// Массив с фотографиями для галереи
const photos = [
  {
    id: 1,
    url: "https://images.unsplash.com/photo-1519681393784-d120267933ba",
    title: "Отпуск на море",
    description: "Наш первый семейный отпуск на Чёрном море. Вода была тёплой, солнце
ласковым.",
    date: "Июль 2023"
  },
  {
    id: 2,
    url: "https://images.unsplash.com/photo-1518837695005-2083093ee35b",
    title: "День рождения внука",
    description: "Петя исполнилось 5 лет! Торт в виде динозавра стал главным сюрпризом.",
    date: "Май 2024"
  },
  {
    id: 3,
    url: "https://images.unsplash.com/photo-1507003211169-0a1dd7228f2d",
    title: "Семейный ужин",
    description: "Все дети и внуки собрались за одним столом. Такие моменты бесценны.",
    date: "Декабрь 2023"
  },
  {
    id: 4,
    url: "https://images.unsplash.com/photo-1519681393784-d120267933ba",
    title: "Сад в цвету",
    description: "Весеннее цветение яблонь. Аромат стоит на весь участок.",
    date: "Апрель 2024"
  },
  {
    id: 5,
    url: "https://images.unsplash.com/photo-1494790108755-2616b612b786",
    title: "Новогодняя ёлка",
    description: "Украшаем ёлку всей семьёй. Внуки вешают шары, которые делали сами.",
    date: "Декабрь 2023"
  },
]
```

```

{
  id: 6,
  url: "https://images.unsplash.com/photo-1517841905240-472988babdf9",
  title: "Первый школьный день",
  description: "Наша младшая внучка Маша пошла в первый класс. Такой взрослой стала!",
  date: "1 сентября 2023"
}
];

// Генерация случайных фотографий с Unsplash (бесплатный сервис)
function getRandomPhotoUrl(index) {
  const sizes = ['300x200', '400x300', '500x400'];
  const size = sizes[Math.floor(Math.random() * sizes.length)];
  const categories = ['nature', 'people', 'family', 'vacation', 'garden'];
  const category = categories[Math.floor(Math.random() * categories.length)];

  return `https://source.unsplash.com/random/${size}?${category}&sig=${index}`;
}

// Создание HTML для одной фотографии
function createPhotoElement(photo, useRandom = false) {
  const imgUrl = useRandom ? getRandomPhotoUrl(photo.id) : photo.url;

  return `
    <div class="photo" data-id="${photo.id}">
      
      <div class="caption">
        <h3>${photo.title}</h3>
        <p>${photo.description}</p>
        <span class="date">${photo.date}</span>
      </div>
    </div>
  `;
}

// Инициализация галереи
function initGallery() {
  const gallery = document.getElementById('gallery');
  gallery.innerHTML = "";

  photos.forEach(photo => {
    gallery.innerHTML += createPhotoElement(photo, true);
  });
}

```

```

// Устанавливаем текущую дату в футере
const now = new Date();
const options = {
    weekday: 'long',
    year: 'numeric',
    month: 'long',
    day: 'numeric'
};
document.getElementById('currentDate').textContent =
    now.toLocaleDateString('ru-RU', options);
}

// Изменение раскладки галереи
function changeLayout(layout) {
    const gallery = document.getElementById('gallery');

    // Удаляем все классы раскладки
    gallery.classList.remove('gallery-row', 'gallery-column', 'gallery-wrap');

    // Добавляем нужный класс
    switch(layout) {
        case 'row':
            gallery.classList.add('gallery-row');
            alert('Галерея переключена в режим "В ряд". Прокручивайте горизонтально!');
            break;
        case 'column':
            gallery.classList.add('gallery-column');
            alert('Галерея переключена в режим "В столбик". Фото теперь одна под другой!');
            break;
        case 'wrap':
            gallery.classList.add('gallery-wrap');
            alert('Галерея переключена в режим "Сетка". Фото автоматически переносятся на
новые строки!');
            break;
    }
}

// Добавление новой фотографии
function addPhoto() {
    const newPhotoId = photos.length + 1;
    const newPhoto = {
        id: newPhotoId,
        url: '',
        title: `Новое фото #${newPhotoId}`,
    }
}

```

```
        description: 'Это фото было добавлено с помощью кнопки "Добавить фото".',
        date: 'Сегодня'
    };

    photos.push(newPhoto);

    const gallery = document.getElementById('gallery');
    gallery.innerHTML += createPhotoElement(newPhoto, true);

    // Плавная прокрутка к новой фотографии
    const newPhotoElement = gallery.lastElementChild;
    newPhotoElement.scrollIntoView({ behavior: 'smooth', block: 'center' });

    // Эффект добавления
    newPhotoElement.style.opacity = '0';
    newPhotoElement.style.transform = 'scale(0.8)';

    setTimeout(() => {
        newPhotoElement.style.transition = 'all 0.5s ease';
        newPhotoElement.style.opacity = '1';
        newPhotoElement.style.transform = 'scale(1)';
    }, 10);

    alert(`Добавлено новое фото! Всего фотографий: ${photos.length}`);
}

// Смена темы
function changeTheme() {
    const body = document.body;
    const currentTheme = body.classList.contains('dark-theme') ? 'dark' : 'light';

    if (currentTheme === 'light') {
        // Включаем тёмную тему
        body.classList.add('dark-theme');
        body.style.background = 'linear-gradient(135deg, #2c3e50 0%, #1a1a2e 100%)';
        document.querySelector('.container').style.background = '#34495e';
        document.querySelector('.container').style.color = '#ecf0f1';
        alert('Включена тёмная тема!');
    } else {
        // Включаем светлую тему
        body.classList.remove('dark-theme');
        body.style.background = 'linear-gradient(135deg, #f5f7fa 0%, #c3cfe2 100%)';
        document.querySelector('.container').style.background = 'white';
        document.querySelector('.container').style.color = '#333';
    }
}
```

```
        alert('Включена светлая тема!');

    }

}

// Добавляем CSS для тёмной темы
const darkThemeCSS = `

.dark-theme .photo {
    background: #2c3e50;
    border-color: #34495e;
}

.dark-theme .caption {
    background: #34495e;
    color: #ecf0f1;
}

.dark-theme .caption h3 {
    color: #ffd700;
}

.dark-theme .explanation {
    background: #34495e;
    color: #ecf0f1;
}

.dark-theme .code-block {
    background: #1a1a2e;
}

`;

// Добавляем стили тёмной темы в head
const style = document.createElement('style');
style.textContent = darkThemeCSS;
document.head.appendChild(style);

// Инициализируем галерею при загрузке страницы
document.addEventListener('DOMContentLoaded', initGallery);
```

---

## Шаг 4: Запускаем и тестируем галерею

### Как запустить:

1. Сохраните все три файла в одной папке:

- index.html
- style.css
- script.js

2. Откройте файл index.html в браузере (двойной клик)

3. Вы увидите готовую фото-галерею!

### Что можно делать с галереей:

#### 1. Тестируем разные режимы:

- «В ряд» — все фото в одну строку (горизонтальная прокрутка)
- «В столбик» — фото одна под другой
- «Сетка» — фото автоматически переносятся (стандартный режим)

#### 2. Интерактивные возможности:

- Наводите мышь на фото — они «оживают»
- Добавляйте новые фото кнопкой
- Меняйте тему (светлая/тёмная)
- Все фото загружаются автоматически

#### 3. Адаптивность:

- На компьютере: 3-4 фото в ряд
- На планшете: 2-3 фото в ряд
- На телефоне: 1 фото в ряд

---

### 🎯 Что мы узнали и создали:

### Волшебство Flexbox в действии:

1. `display: flex` — превращает обычный div в гибкий контейнер

2. flex-wrap: wrap — разрешает перенос на новую строку
3. justify-content: center — центрирует элементы по горизонтали
4. gap: 25px — автоматические промежутки между элементами
5. flex: 1 1 300px — гибкие размеры элементов

## Профессиональные техники:

1. object-fit: cover — красивое кадрирование фотографий
2. **Плавные переходы** — transition для анимации
3. **Адаптивный дизайн** — работает на всех устройствах
4. **Ленивая загрузка** — loading="lazy" для оптимизации

## Интерактивные элементы:

1. Кнопки управления галереей
  2. Динамическое добавление фото
  3. Смена тем оформления
  4. Эффекты при наведении
- 



## Домашнее задание:

### Задание 1: «Моя личная галерея»

1. Замените тестовые фото на свои (можно использовать Unsplash или реальные)
2. Измените подписи под своими фотографиями
3. Добавьте 2-3 своих фото с помощью кнопки «Добавить фото»

### Задание 2: «Эксперимент с настройками»

В файле style.css найдите блок .gallery и поэкспериментируйте:

css

```
.gallery {  
    display: flex;  
    flex-wrap: wrap; /* Попробуйте: nowrap */  
    justify-content: center; /* Попробуйте: flex-start, space-between */
```

gap: 25px; /\* Измените на: 10px, 40px \*/  
}

## Задание 3: «Создай тему»

1. Создайте свою цветовую тему:

- Измените цвета в linear-gradient
- Подберите свою цветовую палитру
- Добавьте свою кнопку для своей темы

## Задание 4: «Галерея достижений»

Создайте новую галерею для:

- Ваших вязаных/связанных работ
  - Выращенных цветов/овощей
  - Внуков в разном возрасте
  - Путешествий
- 



### Советы по улучшению:

1. **Для реальных фото:** Замените <https://source.unsplash.com> на пути к своим файлам
  2. **Для скорости:** Используйте сжатые изображения (можно через [tinypng.com](http://tinypng.com))
  3. **Для доступности:** Всегда заполняйте alt у изображений
  4. **Для красоты:** Добавьте рамки, тени, эффекты на свой вкус
- 

**Дорогой друг, вы только что создали профессиональную фотогалерею!** Теперь вы можете показывать свои фотографии красиво и современно. Flexbox сделал всю сложную работу за вас — элементы действительно «аккуратно выстроились сами».

**Самое главное:** Вы теперь понимаете, как работает `display: flex` и можете применять его для любых элементов на странице — не только для фотографий!

**В следующей главе** мы научимся делать сайт адаптивным — чтобы он красиво смотрелся и на компьютере, и на телефоне!

## ● Глава 12: Как сделать так, чтобы сайтом было удобно пользоваться и на телефоне? (Адаптивность).

- Простое объяснение «медиа-запросов» (@media): Это как разные инструкции для большого телевизора и маленького смартфона.

Дорогой друг, представьте, что у вас есть инструкция по сборке мебели. Для большого шкафа и маленькой тумбочки инструкции будут разные, правда? Так и с сайтами: для большого компьютера и маленького телефона нужны разные «инструкции» по оформлению. Эти «инструкции» и называются **медиа-запросами**!

---

### 🎯 Аналогия: Одна квартира — разные планы расстановки мебели

Представьте, что у вас есть квартира. Вы хотите расставить мебель так, чтобы было удобно:

#### 1. Когда приходят гости (большой экран — компьютер):

Text

БОЛЬШОЙ ДИВАН	ЖУРНАЛЬНЫЙ СТОЛИК
КРЕСЛА	ТЕЛЕВИЗОР ЦВЕТЫ

Много места — можно расставить всё просторно.

#### 2. Когда вы одни (маленький экран — телефон):

Text



Мало места — всё компактно, одно под другим.

### **Медиа-запросы — это как сказать браузеру:**

«Если экран большой — расставь мебель вот так. Если маленький — вот так!»

---

## **Что такое медиа-запросы на самом деле?**

**Медиа-запросы (Media Queries)** — это специальные правила в CSS, которые говорят браузеру:

«Примени эти стили ТОЛЬКО если выполняется определённое условие (например, ширина экрана меньше 768px)»

### **Базовый синтаксис:**

css

```
@media (условие) {  
    /* Стили, которые применяются только при выполнении условия */  
}
```

---

## **Пример из реальной жизни: Текст на разных устройствах**

### **Без медиа-запросов:**

css

```
/* Один размер для всех */
body {
    font-size: 18px; /* На телефоне будет мелко! */
}
```

## С медиа-запросами:

css

```
/* Для компьютеров (большие экраны) */
body {
    font-size: 18px; /* Хорошо читается */
}

/* Для телефонов (маленькие экраны) */
@media (max-width: 768px) {
    body {
        font-size: 16px; /* Чуть меньше, но читаемо */
    }
}

/* Для очень маленьких телефонов */
@media (max-width: 480px) {
    body {
        font-size: 14px; /* Ещё меньше, но всё равно читаемо */
    }
}
```

## Что происходит:

1. На компьютере (ширина больше 768px) — шрифт 18px
  2. На планшете (ширина меньше 768px) — шрифт 16px
  3. На маленьком телефоне (ширина меньше 480px) — шрифт 14px
- 



## Подробный пример с визуализацией

Создайте файл `media-queries.html`:

html

```
<!DOCTYPE html>
<html lang="ru">
```

```
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Медиа-запросы: инструкции для разных экранов</title>
  <style>
    /* ===== ОБЩИЕ СТИЛИ (для всех устройств) ===== */
    * {
      margin: 0;
      padding: 0;
      box-sizing: border-box;
    }

    body {
      font-family: 'Georgia', serif;
      line-height: 1.6;
      color: #333;
      transition: all 0.3s;
    }

    .container {
      max-width: 1200px;
      margin: 0 auto;
      padding: 20px;
    }

    header {
      background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);
      color: white;
      padding: 40px 20px;
      text-align: center;
      border-radius: 15px 15px 0 0;
      margin-bottom: 30px;
    }

    h1 {
      font-size: 2.5rem;
      margin-bottom: 10px;
    }

    /* ===== МЕДИА-ЗАПРОС №1: Для планшетов (max-width: 1024px) ===== */
    @media (max-width: 1024px) {
      /* Стили применяются ТОЛЬКО если ширина экрана ≤ 1024px */

      h1 {
```

```
font-size: 2rem; /* Заголовок становится меньше */
color: #ffd700; /* Меняем цвет на золотой */
}

header {
background: linear-gradient(135deg, #4CAF50 0%, #2E7D32 100%);
/* Меняем градиент на зелёный */
}

.container {
padding: 15px; /* Уменьшаем отступы */
}
}

/* ===== МЕДИА-ЗАПРОС №2: Для телефонов (max-width: 768px) ===== */
@media (max-width: 768px) {
/* Стили применяются ТОЛЬКО если ширина экрана ≤ 768px */

h1 {
font-size: 1.8rem; /* Ещё меньше */
color: #ff6b6b; /* Красный цвет */
}

header {
background: linear-gradient(135deg, #FF9800 0%, #F57C00 100%);
/* Оранжевый градиент */
padding: 25px 15px; /* Меньше отступы */
border-radius: 10px; /* Меньше скругление */
}

.container {
padding: 10px; /* Ещё меньше отступы */
}
}

/* ===== МЕДИА-ЗАПРОС №3: Для очень маленьких телефонов (max-width: 480px) ===== */
@media (max-width: 480px) {
/* Стили применяются ТОЛЬКО если ширина экрана ≤ 480px */

h1 {
font-size: 1.5rem; /* Совсем маленький */
color: #ffffff; /* Белый */
text-shadow: 2px 2px 4px rgba(0,0,0,0.5);
}
}
```

```
header {
    background: linear-gradient(135deg, #E91E63 0%, #C2185B 100%);
    /* Розовый градиент */
    padding: 20px 10px;
    border-radius: 5px;
}

body {
    font-size: 14px; /* Меньше основной текст */
}
}

/* ===== ВИЗУАЛИЗАЦИЯ РАЗНЫХ УСТРОЙСТВ ===== */
.devices {
    display: flex;
    flex-wrap: wrap;
    gap: 30px;
    margin: 40px 0;
    justify-content: center;
}

.device {
    flex: 1;
    min-width: 250px;
    padding: 25px;
    border-radius: 15px;
    text-align: center;
    box-shadow: 0 10px 20px rgba(0,0,0,0.1);
    transition: transform 0.3s;
}

.device:hover {
    transform: translateY(-10px);
}

.desktop {
    background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);
    color: white;
}

.tablet {
    background: linear-gradient(135deg, #4CAF50 0%, #2E7D32 100%);
    color: white;
```

```
}

.mobile {
    background: linear-gradient(135deg, #FF9800 0%, #F57C00 100%);
    color: white;
}

.small-mobile {
    background: linear-gradient(135deg, #E91E63 0%, #C2185B 100%);
    color: white;
}

.device-icon {
    font-size: 60px;
    margin-bottom: 15px;
}

/* ===== ДЕМОНСТРАЦИОННЫЙ БЛОК ===== */
.demo-block {
    background: #f8f9fa;
    padding: 30px;
    border-radius: 15px;
    margin: 30px 0;
    border-left: 6px solid #667eea;
}

.demo-content {
    display: flex;
    gap: 20px;
    margin-top: 20px;
}

.box {
    flex: 1;
    background: #3498db;
    color: white;
    padding: 30px;
    text-align: center;
    border-radius: 10px;
    font-weight: bold;
}

/* Медиа-запрос для демо-блока */
@media (max-width: 768px) {
```

```
.demo-content {  
    flex-direction: column; /* Блоки становятся друг под другом */  
}  
  
.box {  
    margin-bottom: 15px;  
    background: #2ecc71; /* Меняем цвет */  
}  
}  
  
/* ===== ИНДИКАТОР ТЕКУЩЕГО РЕЖИМА ===== */  
.current-mode {  
    position: fixed;  
    top: 20px;  
    right: 20px;  
    background: #2c3e50;  
    color: white;  
    padding: 15px 25px;  
    border-radius: 50px;  
    font-weight: bold;  
    box-shadow: 0 5px 15px rgba(0,0,0,0.3);  
    z-index: 1000;  
    display: flex;  
    align-items: center;  
    gap: 10px;  
}  
  
.mode-icon {  
    font-size: 24px;  
}  
  
/* ===== КОД ПРИМЕРЫ ===== */  
.code-examples {  
    background: #2c3e50;  
    color: #ecf0f1;  
    padding: 30px;  
    border-radius: 15px;  
    margin: 40px 0;  
    font-family: 'Courier New', monospace;  
}  
  
.code-title {  
    color: #2ecc71;  
    margin-bottom: 20px;
```

```
        font-family: 'Georgia', serif;
    }

pre {
    white-space: pre-wrap;
    line-height: 1.5;
}

.comment {
    color: #7f8c8d;
}

.property {
    color: #3498db;
}

.value {
    color: #f1c40f;
}

/* ===== КНОПКА ДЛЯ ТЕСТИРОВАНИЯ ===== */
.test-btn {
    display: block;
    width: 200px;
    margin: 30px auto;
    background: #e74c3c;
    color: white;
    border: none;
    padding: 15px 30px;
    border-radius: 50px;
    font-size: 18px;
    font-weight: bold;
    cursor: pointer;
    transition: all 0.3s;
    text-align: center;
}

.test-btn:hover {
    background: #c0392b;
    transform: scale(1.05);
}

</style>
</head>
<body>
```

```
<!-- Индикатор текущего режима -->
<div class="current-mode" id="modeIndicator">
  <span class="mode-icon" id="modeIcon"></span>
  <span id="modeText">Компьютер</span>
</div>

<div class="container">
  <header>
    <h1>📱 Медиа-запросы: Умные инструкции для разных экранов</h1>
    <p>Попробуйте изменить размер окна браузера и посмотрите, как меняется
страница!</p>
  </header>

  <div class="devices">
    <div class="device desktop">
      <div class="device-icon"></div>
      <h3>Компьютер</h3>
      <p>> 1024px</p>
      <p>Фиолетовый фон, крупный текст</p>
    </div>

    <div class="device tablet">
      <div class="device-icon"></div>
      <h3>Планшет</h3>
      <p>≤ 1024px</p>
      <p>Зелёный фон, золотой текст</p>
    </div>

    <div class="device mobile">
      <div class="device-icon"> </div>
      <h3>Телефон</h3>
      <p>≤ 768px</p>
      <p>Оранжевый фон, красный текст</p>
    </div>

    <div class="device small-mobile">
      <div class="device-icon"> </div>
      <h3>Маленький телефон</h3>
      <p>≤ 480px</p>
      <p>Розовый фон, белый текст с тенью</p>
    </div>
  </div>

  <div class="demo-block">
```

<h2>Демонстрация перестройки макета</h2>  
<p>На компьютере блоки расположены в ряд. На телефоне — становятся друг под другом!</p>

```
<div class="demo-content">  
    <div class="box">Блок 1</div>  
    <div class="box">Блок 2</div>  
    <div class="box">Блок 3</div>  
</div>  
</div>
```

<div class="code-examples">  
 <h2 class="code-title">📝 Как это работает в коде:</h2>

```
<pre><code><span class="comment">/* Базовые стили для всех устройств */</span>
```

```
body {  
    <span class="property">font-size</span>: <span class="value">18px</span>;  
}
```

```
<span class="comment">/* Стили ТОЛЬКО для планшетов (ширина ≤ 1024px) */</span>  
<span class="property">@media </span>(<span class="value">max-width: 1024px</span>) {  
    body {  
        <span class="property">font-size</span>: <span class="value">16px</span>; <span  
class="comment">// Чуть меньше</span>  
    }  
}
```

```
<span class="comment">/* Стили ТОЛЬКО для телефонов (ширина ≤ 768px) */</span>  
<span class="property">@media </span>(<span class="value">max-width: 768px</span>) {  
    body {  
        <span class="property">font-size</span>: <span class="value">14px</span>; <span  
class="comment">// Ещё меньше</span>  
    }  
}
```

```
<span class="comment">/* Стили ТОЛЬКО для маленьких телефонов (ширина ≤ 480px) */</span>  
<span class="property">@media </span>(<span class="value">max-width: 480px</span>) {  
    body {  
        <span class="property">font-size</span>: <span class="value">12px</span>; <span  
class="comment">// Самый маленький</span>  
    }  
}</code></pre>
```

<p style="margin-top: 20px; color: #bdc3c7;">

 Браузер читает код сверху вниз и применяет стили по порядку.

Медиа-запросы как бы говорят: «Если условие верно — примени эти дополнительные стили!»

```
</p>  
</div>
```

```
<div style="text-align: center; margin: 40px 0;">  
    <button class="test-btn" onclick="simulateDevices()">  
         Протестировать все режимы  
    </button>  
    <p><small>Нажмите кнопку, чтобы увидеть, как страница выглядит на разных  
устройствах</small></p>  
</div>
```

```
<div style="background: #fff8e1; padding: 25px; border-radius: 15px; border-left: 6px solid  
#ffb300;">  
    <h2> Ключевые моменты:</h2>  
    <div style="display: flex; flex-wrap: wrap; gap: 20px; margin-top: 20px;">  
        <div style="flex: 1; min-width: 250px; padding: 20px; background: white; border-  
radius: 10px;">  
            <h3> max-width</h3>  
            <p><code>@media (max-width: 768px)</code></p>  
            <p>«Применить стили, если ширина экрана МЕНЬШЕ или РАВНА  
768px»</p>  
        </div>
```

```
<div style="flex: 1; min-width: 250px; padding: 20px; background: white; border-  
radius: 10px;">  
    <h3> min-width</h3>  
    <p><code>@media (min-width: 769px)</code></p>  
    <p>«Применить стили, если ширина экрана БОЛЬШЕ или РАВНА  
769px»</p>  
    </div>
```

```
<div style="flex: 1; min-width: 250px; padding: 20px; background: white; border-  
radius: 10px;">  
    <h3> Комбинирование</h3>  
    <p><code>@media (min-width: 768px) and (max-width: 1024px)</code></p>  
    <p>«Применить стили ТОЛЬКО для планшетов (от 768px до 1024px)»</p>  
    </div>  
    </div>  
</div>
```

```

<script>

    // Функция для обновления индикатора режима
    function updateModeIndicator() {
        const width = window.innerWidth;
        const indicator = document.getElementById('modeIndicator');
        const icon = document.getElementById('modeIcon');
        const text = document.getElementById('modeText');

        if (width > 1024) {
            indicator.style.background = '#667eea';
            icon.textContent = '💻';
            text.textContent = 'Компьютер (>1024px)';
        } else if (width > 768) {
            indicator.style.background = '#4CAF50';
            icon.textContent = '📱';
            text.textContent = 'Планшет (768-1024px)';
        } else if (width > 480) {
            indicator.style.background = '#FF9800';
            icon.textContent = '📲';
            text.textContent = 'Телефон (480-768px)';
        } else {
            indicator.style.background = '#E91E63';
            icon.textContent = '📱';
            text.textContent = 'Маленький телефон (<480px)';
        }
    }

    // Функция для симуляции разных устройств
    function simulateDevices() {
        const widths = [1200, 900, 600, 375]; // Ширины разных устройств
        const deviceNames = ['Компьютер', 'Планшет', 'Телефон', 'Маленький телефон'];

        let currentIndex = 0;

        function simulateNext() {
            if (currentIndex >= widths.length) {
                // Возвращаем исходный размер
                window.resizeTo(window.outerWidth, window.outerHeight);
                updateModeIndicator();
                alert('Тестирование завершено!');
                return;
            }
        }
    }

    // Изменяем размер окна (только для демонстрации!)

```

```

// В реальном браузере это может не работать из-за политики безопасности
try {
    window.resizeTo(widths[currentIndex] + 50, 800);
} catch(e) {
    // Если не получается изменить размер, просто показываем сообщение
}

updateModeIndicator();
alert(`Сейчас симулируется: ${deviceNames[currentIndex]} ($
${widths[currentIndex]}px шириной)\n\nВ реальном проекте просто измените размер окна браузера!
`);

currentIndex++;
setTimeout(simulateNext, 3000);
}

simulateNext();
}

// Обновляем индикатор при изменении размера окна
window.addEventListener('resize', updateModeIndicator);

// Инициализируем индикатор при загрузке
document.addEventListener('DOMContentLoaded', updateModeIndicator);

// Альтернативный способ тестирования для браузеров, которые не позволяют resizeTo
function showMediaQueryInfo() {
    const width = window.innerWidth;
    let info = `Текущая ширина окна: ${width}px\n\n`;

    if (width > 1024) {
        info += '✓ Активны: Базовые стили\n';
        info += '✗ Не активны: Медиа-запросы для планшетов, телефонов и маленьких
телефонов';
    } else if (width > 768) {
        info += '✓ Активны: Базовые стили + медиа-запрос для планшетов\n';
        info += '✗ Не активны: Медиа-запросы для телефонов и маленьких телефонов';
    } else if (width > 480) {
        info += '✓ Активны: Базовые стили + медиа-запрос для планшетов + для
телефонов\n';
        info += '✗ Не активны: Медиа-запрос для маленьких телефонов';
    } else {
        info += '✓ Активны: ВСЕ медиа-запросы\n';
        info += 'Все стили применены!';
    }
}

```

```
        }

        alert(info);
    }

    // Добавляем кнопку для альтернативного тестирования
    const testBtn = document.querySelector('.test-btn');
    testBtn.onclick = showMediaQueryInfo;
</script>
</body>
</html>
```

---

## ⌚ Основные типы медиа-запросов:

### 1. По ширине (самые частые):

css

```
/* Для очень больших экранов (телевизоры, мониторы 4K) */
@media (min-width: 1200px) { ... }

/* Для компьютеров и ноутбуков */
@media (min-width: 992px) { ... }

/* Для планшетов (альбомная ориентация) */
@media (max-width: 991px) { ... }

/* Для планшетов (портретная ориентация) и больших телефонов */
@media (max-width: 768px) { ... }

/* Для телефонов */
@media (max-width: 576px) { ... }
```

### 2. По высоте (редко, но бывает нужно):

css

```
/* Если экран очень высокий */
@media (min-height: 800px) { ... }

/* Если экран очень низкий */
@media (max-height: 600px) { ... }
```

### 3. По ориентации:

css

```
/* Альбомная (горизонтальная) */
@media (orientation: landscape) { ... }

/* Портретная (вертикальная) */
@media (orientation: portrait) { ... }
```

---



## Реальный пример: Адаптивное меню

### Без медиа-запросов:

css

```
.menu {
    display: flex;
    gap: 30px;
}

/* На телефоне кнопки будут слишком маленькими или вылезут за край */
```

### С медиа-запросами:

css

```
/* Для компьютеров */
.menu {
    display: flex;
    gap: 30px;
}

/* Для телефонов */
@media (max-width: 768px) {
    .menu {
        flex-direction: column; /* Кнопки друг под другом */
        gap: 10px; /* Меньшие промежутки */
        font-size: 14px; /* Меньше текст */
    }
}
```

---

## Практическое задание:

### **Задание 1: «Адаптивная визитка»**

1. Создайте свою визитную карточку
2. Для компьютера: фото слева, текст справа
3. Для телефона: фото сверху, текст снизу
4. Используйте медиа-запрос @media (max-width: 768px)

### **Задание 2: «Умный текст»**

1. Создайте страницу с длинным текстом
2. Для компьютера: ширина текста 800px, шрифт 18px
3. Для планшета: ширина 600px, шрифт 16px
4. Для телефона: ширина 100%, шрифт 14px, больше межстрочное расстояние

### **Задание 3: «Галерея из практикума 7»**

Вернитесь к галерее из практикума 7 и добавьте:

1. На компьютере: 4 фото в ряд
2. На планшете: 2 фото в ряд
3. На телефоне: 1 фото в ряд

### **Задание 4: «Кнопка-хамелеон»**

Создайте кнопку, которая меняет цвет в зависимости от размера экрана:

css

```
.btn {  
    padding: 15px 30px;  
    color: white;  
}  
  
/* Зелёная на компьютере */  
@media (min-width: 1024px) { .btn { background: green; } }  
  
/* Синяя на планшете */  
@media (max-width: 1023px) { .btn { background: blue; } }  
  
/* Красная на телефоне */  
@media (max-width: 768px) { .btn { background: red; } }
```



## Советы по использованию медиа-запросов:

### 1. Начинайте с мобильной версии (Mobile First):

css

```
/* Сначала стили для телефона */
```

```
body { font-size: 14px; }
```

```
/* Потом для больших экранов */
```

```
@media (min-width: 768px) {
```

```
    body { font-size: 16px; }
```

```
}
```

### 2. Используйте логические операторы:

css

```
/* И (and) - оба условия должны быть true */
```

```
@media (min-width: 768px) and (max-width: 1024px) { ... }
```

```
/* Или (,) - хотя бы одно условие должно быть true */
```

```
@media (max-width: 768px), (orientation: portrait) { ... }
```

```
/* Не (not) - условие НЕ должно быть true */
```

```
@media not (max-width: 600px) { ... }
```

### 3. Стандартные брейкпоинты (точки перелома):

- 576px — маленькие телефоны
- 768px — планшеты/телефоны (вертикально)
- 992px — планшеты (горизонтально)/небольшие ноутбуки
- 1200px — компьютеры и ноутбуки
- 1400px — большие мониторы

### 4. Тестируйте в браузере:

- F12 → Toggle Device Toolbar (Ctrl+Shift+M)
  - Выбирайте разные устройства
  - Меняйте ориентацию
-

## **Что такое «Адаптивный дизайн» простыми словами:**

**Адаптивный дизайн** — это когда ваш сайт как **умная одежда**:

- На большом человеке (компьютер) — просторный костюм
- На среднем (планшет) — удобные джинсы и футболка
- На маленьком (телефон) — компактная спортивная форма

**Медиа-запросы** — это **швейная машинка**, которая перешивает одежду под нужный размер!

---

**Дорогой друг, теперь вы знаете секрет, как сделать сайт удобным на любом устройстве!** Медиа-запросы — это ваш волшебный инструмент, который говорит браузеру: «На большом экране — делай так, на маленьком — делай этак!»

**Самое главное:** Вы можете сделать один сайт, который будет красиво смотреться и на компьютере внука, и на вашем телефоне, и на планшете жены!

**В следующей (заключительной) главе** мы подведём итоги всего, что вы узнали, и покажем, куда двигаться дальше в вашем путешествии в мир веб-дизайна!

## ■ Пример: Меняем размер шрифта и расположение элементов на маленьком экране.

Дорогой друг, давайте представим реальную ситуацию: вы создали красивый сайт о своём хобби на компьютере. Всё прекрасно: крупные заголовки, фотографии рядом с текстом, удобное меню. Но когда вы открываете его на телефоне... о ужас! Текст такой мелкий, что нужно увеличивать, фотографии не помещаются, а меню «уезжает» за край экрана.

Именно для этого мы и используем медиа-запросы! Давайте создадим конкретный, работающий пример и посмотрим, как волшебным образом страница адаптируется под маленький экран.

---

## ■ Полный рабочий пример: Страница садовода

Создайте файл adaptivnaya-stranica.html:

```
html
<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Сад моей мечты - адаптивная страница</title>
    <style>
        /* ===== ОБЩИЕ СТИЛИ (работают на всех устройствах) ===== */
        * {
            margin: 0;
            padding: 0;
            box-sizing: border-box;
        }

        body {
            font-family: 'Georgia', serif;
            line-height: 1.6;
            color: #333;
            background-color: #f0ffff; /* Светло-зелёный фон */
            padding: 20px;
        }
    
```

```
}

.container {
    max-width: 1200px;
    margin: 0 auto;
    background: white;
    border-radius: 20px;
    box-shadow: 0 10px 30px rgba(0,0,0,0.1);
    overflow: hidden;
}

/* ===== ШАПКА САЙТА ===== */
header {
    background: linear-gradient(135deg, #2e8b57 0%, #228b22 100%);
    color: white;
    padding: 40px;
    text-align: center;
    border-bottom: 8px solid #98fb98;
}

h1 {
    font-size: 48px; /* ОЧЕНЬ КРУПНЫЙ на компьютере */
    margin-bottom: 15px;
    text-shadow: 2px 2px 4px rgba(0,0,0,0.3);
}

.subtitle {
    font-size: 24px; /* Крупный подзаголовок */
    font-style: italic;
    opacity: 0.9;
}

/* ===== НАВИГАЦИОННОЕ МЕНЮ ===== */
nav {
    background: #f8ffff;
    padding: 20px;
    border-bottom: 3px solid #98fb98;
}

.menu {
    display: flex; /* Кнопки в ряд на компьютере */
    justify-content: center;
    list-style: none;
    gap: 40px; /* Большие промежутки */
}
```

```
}

.menu a {
    text-decoration: none;
    color: #2e8b57;
    font-size: 20px; /* Крупный текст в меню */
    font-weight: bold;
    padding: 12px 24px;
    border-radius: 30px;
    transition: all 0.3s;
    border: 2px solid transparent;
}

.menu a:hover {
    background: #98fb98;
    color: #006400;
    border-color: #2e8b57;
    transform: translateY(-3px);
}

/* ===== ОСНОВНОЕ СОДЕРЖАНИЕ ===== */
.content {
    display: flex; /* Фото и текст рядом на компьютере */
    padding: 40px;
    gap: 50px; /* Большой промежуток между колонками */
}

.main-photo {
    flex: 1;
}

.main-photo img {
    width: 100%;
    height: 400px;
    object-fit: cover;
    border-radius: 15px;
    box-shadow: 0 10px 20px rgba(0,0,0,0.2);
    border: 5px solid #98fb98;
}

.description {
    flex: 1;
}
```

```
.description h2 {  
    color: #2e8b57;  
    font-size: 36px; /* Крупный заголовок */  
    margin-bottom: 25px;  
    border-bottom: 3px solid #98fb98;  
    padding-bottom: 10px;  
}  
  
.description p {  
    font-size: 20px; /* Крупный текст для чтения */  
    line-height: 1.8;  
    margin-bottom: 20px;  
    color: #444;  
}  
  
.highlight {  
    background: #f0ffff;  
    padding: 25px;  
    border-radius: 15px;  
    border-left: 8px solid #2e8b57;  
    font-size: 22px; /* Особо крупный текст */  
    font-style: italic;  
    color: #006400;  
    margin: 30px 0;  
}  
  
/* ===== ГАЛЕРЕЯ ФОТО ===== */  
.gallery {  
    padding: 40px;  
    background: #f8ffff;  
    border-top: 3px solid #98fb98;  
}  
  
.gallery h2 {  
    text-align: center;  
    color: #2e8b57;  
    font-size: 36px;  
    margin-bottom: 40px;  
}  
  
.photos {  
    display: flex; /* Все фото в один ряд на компьютере */  
    gap: 30px; /* Большие промежутки */  
    justify-content: center;
```

```
}

.photo-item {
    flex: 1;
    max-width: 300px;
}

.photo-item img {
    width: 100%;
    height: 250px;
    object-fit: cover;
    border-radius: 10px;
    border: 3px solid #98fb98;
}

.photo-item p {
    text-align: center;
    margin-top: 15px;
    font-size: 18px;
    color: #2e8b57;
    font-weight: bold;
}

/* ===== КОНТАКТЫ ===== */

.contacts {
    padding: 40px;
    background: white;
    display: flex; /* Конакты и карта рядом */
    gap: 50px;
}

.contact-info {
    flex: 1;
}

.contact-info h2 {
    color: #2e8b57;
    font-size: 36px;
    margin-bottom: 25px;
}

.contact-info p {
    font-size: 20px;
    margin-bottom: 15px;
}
```

```
}

.map {
    flex: 1;
    background: #e8f5e9;
    border-radius: 15px;
    display: flex;
    align-items: center;
    justify-content: center;
    font-size: 24px;
    color: #2e8b57;
    border: 3px dashed #98fb98;
}

/* ===== ФУТЭР ===== */

footer {
    background: #2e8b57;
    color: white;
    text-align: center;
    padding: 30px;
    font-size: 18px;
    border-top: 8px solid #98fb98;
}

/* ===== ★★★★ ВОЛШЕБНЫЕ МЕДИА-ЗАПРОСЫ ★★★★ ===== */

/* ===== ЗАПРОС 1: Для планшетов (ширина ≤ 1024px) ===== */

@media (max-width: 1024px) {
    /* Немного уменьшаем размеры для планшетов */

    h1 {
        font-size: 40px; /* Чуть меньше заголовок */
    }

    .subtitle {
        font-size: 22px;
    }

    .menu {
        gap: 20px; /* Меньше промежутки между кнопками */
    }

    .menu a {
        font-size: 18px;
    }
}
```

```
padding: 10px 20px;
}

.content {
    gap: 30px; /* Меньше промежуток между фото и текстом */
}

.description h2 {
    font-size: 32px;
}

.description p {
    font-size: 18px;
}

.highlight {
    font-size: 20px;
}

.photos {
    gap: 20px; /* Фото ближе друг к другу */
}

/*
===== ЗАПРОС 2: Для телефонов (ширина ≤ 768px) =====
@media (max-width: 768px) {
    /* ЗНАЧИТЕЛЬНЫЕ ИЗМЕНЕНИЯ ДЛЯ ТЕЛЕФОНОВ!
    */

    /* 1. УМЕНЬШАЕМ ВСЕ РАЗМЕРЫ ШРИФТОВ */
    h1 {
        font-size: 32px !important; /* Намного меньше!
        padding: 0 10px;
    }

    .subtitle {
        font-size: 18px !important;
    }

    /* 2. МЕНЯЕМ РАСПОЛОЖЕНИЕ МЕНЮ */
    .menu {
        flex-direction: column; /* Кнопки теперь ВЕРТИКАЛЬНО!
        gap: 10px; /* Маленькие промежутки
        align-items: center;
    }
}
```

```
.menu a {  
    width: 90%; /* Кнопки почти на всю ширину */  
    text-align: center;  
    font-size: 18px;  
    padding: 15px;  
    margin-bottom: 5px;  
}  
  
/* 3. МЕНЯЕМ РАСПОЛОЖЕНИЕ КОНТЕНТА */  
.content {  
    flex-direction: column; /* Фото и текст теперь друг под другом! */  
    gap: 30px;  
    padding: 25px;  
}  
  
.main-photo img {  
    height: 300px; /* Меньшая высота фото */  
}  
  
/* 4. ЕЩЁ МЕНЬШЕ ШРИФТЫ */  
.description h2 {  
    font-size: 28px;  
    text-align: center;  
}  
  
.description p {  
    font-size: 17px; /* Удобно для чтения на телефоне */  
    line-height: 1.7;  
}  
  
.highlight {  
    font-size: 18px;  
    padding: 20px;  
}  
  
/* 5. МЕНЯЕМ ГАЛЕРЕЮ */  
.gallery {  
    padding: 25px;  
}  
  
.gallery h2 {  
    font-size: 28px;  
}
```

```
.photos {
    flex-direction: column; /* Фото теперь в столбик! */
    align-items: center;
    gap: 25px;
}

.photo-item {
    max-width: 100%; /* Фото на всю ширину */
}

.photo-item img {
    height: 200px;
}

/* 6. МЕНЯЕМ КОНТАКТЫ */
.contacts {
    flex-direction: column; /* Контакты и карта друг под другом */
    gap: 30px;
    padding: 25px;
}

.contact-info h2 {
    font-size: 28px;
    text-align: center;
}

.contact-info p {
    font-size: 18px;
}

.map {
    height: 200px; /* Меньшая карта */
    font-size: 20px;
}

/* 7. ФУТЕР */
footer {
    padding: 20px;
    font-size: 16px;
}
```

/\* ===== ЗАПРОС 3: Для очень маленьких телефонов (ширина ≤ 480px) ===== \*/

```
@media (max-width: 480px) {  
    /* ЕЩЁ БОЛЬШЕ АДАПТАЦИИ ДЛЯ МАЛЕНЬКИХ ЭКРАНОВ */  
  
    body {  
        padding: 10px; /* Минимальные отступы по краям */  
    }  
  
    header {  
        padding: 25px 15px; /* Меньше отступы в шапке */  
    }  
  
    h1 {  
        font-size: 28px !important; /* Минимальный размер */  
    }  
  
.subtitle {  
    font-size: 16px !important;  
}  
  
.menu a {  
    font-size: 16px; /* Ещё меньше кнопки меню */  
    padding: 12px;  
}  
  
.content, .gallery, .contacts {  
    padding: 15px; /* Минимальные внутренние отступы */  
}  
  
.main-photo img {  
    height: 200px; /* Минимальная высота фото */  
    border-width: 3px; /* Тонкая рамка */  
}  
  
.description h2 {  
    font-size: 24px;  
}  
  
.description p {  
    font-size: 16px; /* Оптимальный размер для маленького экрана */  
    line-height: 1.6;  
}  
  
.highlight {  
    font-size: 16px;
```

```
padding: 15px;
}

.gallery h2 {
    font-size: 24px;
}

.photo-item img {
    height: 150px; /* Маленькие фото в галерее */
}

.contact-info h2 {
    font-size: 24px;
}

.contact-info p {
    font-size: 16px;
}

.map {
    height: 150px;
    font-size: 18px;
}

footer {
    font-size: 14px;
    padding: 15px;
}

/*
===== ИНФОРМАЦИОННАЯ ПАНЕЛЬ =====
*/

.info-panel {
    position: fixed;
    bottom: 20px;
    right: 20px;
    background: #2c3e50;
    color: white;
    padding: 15px 25px;
    border-radius: 10px;
    font-family: 'Courier New', monospace;
    box-shadow: 0 5px 15px rgba(0,0,0,0.3);
    z-index: 1000;
    max-width: 300px;
}
```

```
.info-title {
    color: #ffd700;
    font-weight: bold;
    margin-bottom: 5px;
}

.current-size {
    color: #2ecc71;
    font-weight: bold;
}

.test-buttons {
    display: flex;
    gap: 10px;
    margin-top: 20px;
    justify-content: center;
}

.test-btn {
    background: #3498db;
    color: white;
    border: none;
    padding: 8px 15px;
    border-radius: 5px;
    cursor: pointer;
    font-size: 14px;
}

.test-btn:hover {
    background: #2980b9;
}

</style>
</head>
<body>
    
    <div class="info-panel" id="infoPanel">
        <div class="info-title">  Текущий размер экрана:</div>
        <div class="current-size" id="screenSize">Загрузка...</div>
        <div id="activeQueries">Активные медиа-запросы: нет</div>

        <div class="test-buttons">
            <button class="test-btn" onclick="showChanges()">Что изменилось?</button>
            <button class="test-btn" onclick="toggleInfo()">Скрыть</button>
        </div>
    </div>
</body>
```

```
</div>
</div>

<div class="container">
    <!-- ШАПКА -->
    <header>
        <h1>🌿 Сад моей мечты</h1>
        <p class="subtitle">История создания райского уголка на даче</p>
    </header>

    <!-- МЕНЮ -->
    <nav>
        <ul class="menu">
            <li><a href="#"><i class="fas fa-home"></i> Главная</a></li>
            <li><a href="#"><i class="fas fa-seedling"></i> Мои растения</a></li>
            <li><a href="#"><i class="fas fa-calendar"></i> Сезонные работы</a></li>
            <li><a href="#"><i class="fas fa-camera"></i> Фотогалерея</a></li>
            <li><a href="#"><i class="fas fa-envelope"></i> Контакты</a></li>
        </ul>
    </nav>

    <!-- ОСНОВНОЕ СОДЕРЖАНИЕ -->
    <div class="content">
        <div class="main-photo">
            
        </div>

        <div class="description">
            <h2>Как всё начиналось...</h2>
            <p>Когда мы купили этот участок 10 лет назад, здесь была только голая земля и несколько старых деревьев. Но я видела потенциал! Каждый год мы с мужем что-то улучшали, сажали новые растения, строили дорожки.</p>
            <p>Сейчас наш сад — это место, куда приезжают не только дети и внуки, но и соседи полюбоваться цветением. Особенно красиво весной, когда одновременно цветут яблони, сирень и тюльпаны.</p>
        <div class="highlight">
            «Сад — это не место, а путь. Не цель, а путешествие. Каждый день в нём — это новая страница истории.»
        </div>
    </div>
```

<p>На этой странице я хочу поделиться своим опытом, фотографиями и, конечно, ошибками, которые совершила. Надеюсь, мой опыт поможет другим начинающим садоводам!</p>

</div>

</div>

<!-- ГАЛЕРЕЯ -->

<div class="gallery">

<h2>Мои любимые уголки сада</h2>

<div class="photos">

<div class="photo-item">



<p>Розарий у беседки</p>

</div>

<div class="photo-item">



<p>Органический огород</p>

</div>

<div class="photo-item">



<p>Маленький пруд с кувшинками</p>

</div>

</div>

</div>

<!-- КОНТАКТЫ -->

<div class="contacts">

<div class="contact-info">

<h2>Приезжайте в гости!</h2>

<p><strong>📍 Адрес:</strong> Ленинградская область, дачный посёлок

"Солнечный", участок 42</p>

<p><strong>📞 Телефон:</strong> +7 (921) 123-45-67</p>

<p><strong>✉️ Email:</strong> moy-sad@example.com</p>

<p><strong>☀️ Лучшее время для визита:</strong> С мая по сентябрь, по выходным с 10:00 до 18:00</p>

<p>Обязательно предварительно позвоните, чтобы я могла подготовиться к встрече!</p>

</div>

```
<div class="map">
    [ Здесь будет карта проезда ]
</div>
</div>

<!-- ФУТЕР -->
<footer>
    <p>© 2024 "Сад моей мечты". Все фотографии сделаны лично мной.</p>
    <p>Сайт создан с помощью HTML и CSS. Адаптирован для всех устройств!</p>
</footer>
</div>

<script>
    // Функция для обновления информации о размере экрана
    function updateScreenInfo() {
        const width = window.innerWidth;
        const height = window.innerHeight;
        const screenSizeElement = document.getElementById('screenSize');
        const activeQueriesElement = document.getElementById('activeQueries');

        // Показываем текущий размер
        screenSizeElement.textContent = `${width}px x ${height}px`;

        // Определяем, какие медиа-запросы активны
        let activeQueries = [];

        if (width <= 480) {
            activeQueries.push('Очень маленький телефон (<=480px)');
        } else if (width <= 768) {
            activeQueries.push('Телефон (<=768px)');
        } else if (width <= 1024) {
            activeQueries.push('Планшет (<=1024px)');
        } else {
            activeQueries.push('Компьютер (>1024px)');
        }

        activeQueriesElement.textContent = `Активные медиа-запросы: ${activeQueries.join(',')}`;
    }

    // Меняем цвет панели в зависимости от размера
    const infoPanel = document.getElementById('infoPanel');
    if (width <= 480) {
        infoPanel.style.background = '#E91E63';
    }
}
```

```

} else if (width <= 768) {
    infoPanel.style.background = '#FF9800';
} else if (width <= 1024) {
    infoPanel.style.background = '#4CAF50';
} else {
    infoPanel.style.background = '#2c3e50';
}

}

// Функция для показа изменений
function showChanges() {
    const width = window.innerWidth;
    let message = `Текущая ширина: ${width}px\n\n`;

    if (width > 1024) {
        message += "✓ Режим КОМПЬЮТЕР:\n";
        message += "• Заголовок: 48px\n";
        message += "• Меню: кнопки в ряд\n";
        message += "• Контент: фото и текст рядом\n";
        message += "• Галерея: 3 фото в ряд\n";
        message += "• Контакты: текст и карта рядом\n";
    } else if (width > 768) {
        message += "✓ Режим ПЛАНШЕТ:\n";
        message += "• Заголовок: 40px\n";
        message += "• Меню: кнопки в ряд, но ближе\n";
        message += "• Контент: фото и текст рядом\n";
        message += "• Галерея: 3 фото в ряд\n";
        message += "• Контакты: текст и карта рядом\n";
    } else if (width > 480) {
        message += "✓ Режим ТЕЛЕФОН:\n";
        message += "• Заголовок: 32px (мельче!)\n";
        message += "• Меню: кнопки в столбик\n";
        message += "• Контент: фото и текст друг под другом!\n";
        message += "• Галерея: фото в столбик!\n";
        message += "• Контакты: текст и карта друг под другом!\n";
    } else {
        message += "✓ Режим МАЛЕНЬКИЙ ТЕЛЕФОН:\n";
        message += "• Заголовок: 28px (ещё мельче!)\n";
        message += "• Меню: кнопки в столбик, шире\n";
        message += "• Контент: фото меньше, текст мельче\n";
        message += "• Галерея: маленькие фото в столбик\n";
        message += "• Контакты: компактное расположение\n";
    }
}

```

```
message += "\n 📱 Попробуйте изменить размер окна браузера!";
alert(message);
}

// Функция для скрытия/показа панели
function toggleInfo() {
    const panel = document.getElementById('infoPanel');
    if (panel.style.display === 'none') {
        panel.style.display = 'block';
        event.target.textContent = 'Скрыть';
    } else {
        panel.style.display = 'none';
        event.target.textContent = 'Показать';
    }
}

// Обновляем информацию при изменении размера окна
window.addEventListener('resize', updateScreenInfo);

// Инициализация при загрузке
document.addEventListener('DOMContentLoaded', function() {
    updateScreenInfo();

    // Добавляем иконки Font Awesome
    const faLink = document.createElement('link');
    faLink.rel = 'stylesheet';
    faLink.href = 'https://cdnjs.cloudflare.com/ajax/libs/font-awesome/6.4.0/css/all.min.css';
    document.head.appendChild(faLink);
});

// Функция для симуляции разных устройств (альтернатива resize)
function simulateDevice(device) {
    let message = "";

    switch(device) {
        case 'computer':
            message = "💻 РЕЖИМ КОМПЬЮТЕРА:\n\n";
            message += "• Крупные заголовки (48px)\n";
            message += "• Меню горизонтальное\n";
            message += "• Фото и текст рядом\n";
            message += "• 3 фото в ряд в галерее\n";
            message += "• Широкие отступы\n";
            break;
    }
}
```

```

        case 'tablet':
            message = "📱 РЕЖИМ ПЛАНШЕТА:\n\n";
            message += "• Средние заголовки (40px)\n";
            message += "• Меню ещё горизонтальное\n";
            message += "• Фото и текст ещё рядом\n";
            message += "• 3 фото в ряд, но ближе\n";
            message += "• Средние отступы\n";
            break;

        case 'phone':
            message = "+📱 РЕЖИМ ТЕЛЕФОНА:\n\n";
            message += "• Меньшие заголовки (32px)\n";
            message += "• Меню стало вертикальным!\n";
            message += "• Фото и текст теперь друг под другом!\n";
            message += "• Фото в галерее теперь в столбик!\n";
            message += "• Контакты тоже вертикально!\n";
            break;

        case 'small-phone':
            message = "{📱} РЕЖИМ МАЛЕНЬКОГО ТЕЛЕФОНА:\n\n";
            message += "• Самые маленькие заголовки (28px)\n";
            message += "• Минимальные отступы\n";
            message += "• Компактное расположение всего\n";
            message += "• Маленькие фото\n";
            message += "• Очень плотная вёрстка\n";
            break;
    }

    message += "\nЧтобы увидеть это в действии, измените размер окна браузера!";
    alert(message);
}

// Добавляем кнопки для симуляции
document.addEventListener('DOMContentLoaded', function() {
    const testButtons = `
        <div style="text-align: center; margin: 20px;">
            <h3>🎮 Тестируйте режимы:</h3>
            <button onclick="simulateDevice('computer')" style="margin: 5px; padding: 10px 15px; background: #667eee; color: white; border: none; border-radius: 5px; cursor: pointer;">💻
                Компьютер
            </button>
            <button onclick="simulateDevice('tablet')" style="margin: 5px; padding: 10px 15px; background: #4CAF50; color: white; border: none; border-radius: 5px; cursor: pointer;">📱
                Планшет
            </button>
    `;
})
```

```
<button onclick="simulateDevice('phone')" style="margin: 5px; padding: 10px  
15px; background: #FF9800; color: white; border: none; border-radius: 5px; cursor: pointer;"> Телефон</button>  
                  <button onclick="simulateDevice('small-phone')" style="margin: 5px; padding:  
10px 15px; background: #E91E63; color: white; border: none; border-radius: 5px; cursor: pointer;"> Маленький телефон</button>  
        </div>  
        `;  
  
    // Добавляем кнопки перед футером  
    const container = document.querySelector('.container');  
    const buttonsDiv = document.createElement('div');  
    buttonsDiv.innerHTML = testButtons;  
    container.insertBefore(buttonsDiv, document.querySelector('footer'));  
};  
</script>  
</body>  
</html>
```

---

## **Что конкретно меняется на маленьком экране:**

### **1. Размеры шрифтов (самое важное!):**

#### **На компьютере:**

- Заголовок: 48px (очень крупный)
- Основной текст: 20px (удобно читать с расстояния)
- Меню: 20px (крупные кнопки)

#### **На телефоне:**

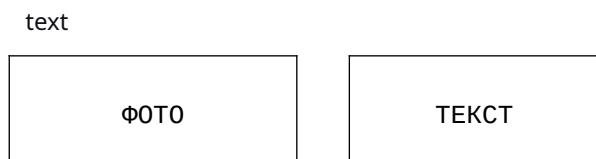
- Заголовок: 32px (в 1.5 раза меньше!)
- Основной текст: 17px (оптимально для маленького экрана)
- Меню: 18px (компактные кнопки)

#### **На очень маленьком телефоне:**

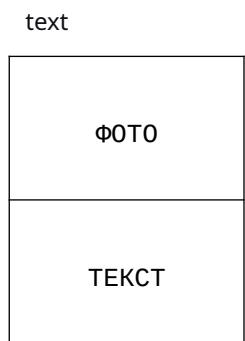
- Заголовок: 28px (минимальный читаемый размер)
- Основной текст: 16px (стандарт для мобильных)

## **2. Расположение элементов (кардинальные изменения!):**

**На компьютере (рядом):**



**На телефоне (друг под другом):**



## **3. Навигационное меню:**

**На компьютере:** Горизонтально, кнопки в ряд с большими промежутками

**На телефоне:** Вертикально, кнопки друг под другом, занимают почти всю ширину

## **4. Галерея фотографий:**

**На компьютере:** 3 фото в один ряд

**На телефоне:** Фото в столбик, одно под другим

## 5. Отступы и промежутки:

**На компьютере:** Большие отступы (40-50px)

**На телефоне:** Меньшие отступы (15-25px)

**На маленьком телефоне:** Минимальные отступы (10-15px)

---

## Как это работает технически:

### Ключевые CSS-свойства, которые меняем:

css

```
/* 1. Меняем размер шрифта */  
font-size: 48px; /* Для компьютера */  
font-size: 32px; /* Для телефона (в медиа-запросе) */  
  
/* 2. Меняем направление flex-контейнера */  
flex-direction: row; /* Для компьютера (рядом) */  
flex-direction: column; /* Для телефона (столбик) */  
  
/* 3. Меняем ширину элементов */  
width: 300px; /* Для компьютера (фиксированная) */  
width: 100%; /* Для телефона (на всю ширину) */  
  
/* 4. Меняем отступы */  
padding: 40px; /* Для компьютера */  
padding: 20px; /* Для телефона */
```

### Последовательность медиа-запросов:

css

```
/* 1. Сначала стили для компьютера (базовые) */  
  
/* 2. Потом для планшетов */  
@media (max-width: 1024px) { ... }  
  
/* 3. Потом для телефонов */  
@media (max-width: 768px) { ... }
```

```
/* 4. Потом для очень маленьких телефонов */
@media (max-width: 480px) { ... }
```

**Важно:** Браузер читает CSS сверху вниз и применяет последнее подходящее правило!

---



## Практическое задание:

### Задание 1: «Протестируйте изменения»

1. Откройте файл `adaptivnaya-stranica.html` в браузере
2. Измените размер окна (перетащите край окна)
3. Наблюдайте, как меняются:
  - Размер шрифтов
  - Расположение элементов
  - Размеры фотографий
  - Расположение меню

### Задание 2: «Добавьте свой раздел»

1. Добавьте новый раздел "Мои советы" на страницу
2. Сделайте его адаптивным:
  - На компьютере: 2 колонки
  - На телефоне: 1 колонка
  - На маленьком телефоне: меньшие шрифты

### Задание 3: «Создайте адаптивную таблицу»

html

```
<div class="pricing">
    <div class="plan">Базовый</div>
    <div class="plan">Стандартный</div>
    <div class="plan">Премиум</div>
</div>
```

css

```
/* Для компьютера: планы в ряд */
.pricing { display: flex; }

/* Для телефона: планы в столбик */
@media (max-width: 768px) {
    .pricing { flex-direction: column; }
    .plan { width: 100%; margin-bottom: 20px; }
}
```

## Задание 4: «Адаптивные изображения»

Создайте изображение, которое:

1. На компьютере: 800px шириной
  2. На планшете: 600px шириной
  3. На телефоне: 100% ширины (но не больше 400px)
  4. На маленьком телефоне: 100% ширины
- 



### Советы по созданию адаптивных страниц:

1. **Всегда начинайте с мобильной версии** — проще увеличивать, чем уменьшать
  2. **Используйте относительные единицы** (% , vw, rem) вместо фиксированных (px)
  3. **Тестируйте на реальных устройствах** или в инструментах разработчика (F12)
  4. **Проверяйте читаемость** — текст не должен быть меньше 14px
  5. **Учитывайте touch-интерфейс** — кнопки должны быть не менее 44x44px для пальцев
- 

## 🎯 Итог: Что вы научились делать?

- ✓ Изменять размеры шрифтов в зависимости от размера экрана
- ✓ Перестраивать макет (ряд → столбик и наоборот)
- ✓ Адаптировать изображения и галереи
- ✓ Создавать адаптивное меню

- Использовать медиа-запросы @media (max-width: ...)
- Тестировать адаптивность на разных размерах экрана

**Дорогой друг, теперь вы умеете создавать сайты, которые будут красиво смотреться на любом устройстве!** Ваши внуки смогут удобно смотреть ваш сайт на телефоне, вы — на планшете, а соседи — на компьютере.

**Это последняя техническая глава нашего учебника!** В следующей, заключительной главе, мы подведём итоги всего, что вы узнали, и покажем, куда двигаться дальше в мире веб-дизайна.

## **Заключение: Ваше творчество только начинается!**

### **● Подводим итоги: Что мы теперь умеем?**

Дорогой друг, наше путешествие подходит к концу, но ваше творчество только начинается! Давайте оглянемся назад и посмотрим, каким огромным путём мы прошли вместе. Вы — уже не просто пользователь интернета, а настоящий **творец цифрового пространства!**

---

### **Часть 1: Мы освоили кисти и краски (основы CSS)**

#### **Глава 1: Поняли, что такое CSS**

- **Узнали:** CSS — это «обои и мебель» для голых HTML-стен
- **Научились:** Подключать стили тремя способами:
  1. Прямо в теге (как записка на холодильник) — `style="color: red;"`
  2. В шапке страницы (как инструкция для комнаты) — `<style>...</style>`
  3. В отдельном файле (как книга рецептов) — `<link rel="stylesheet" href="style.css">`
- **Создали:** Первую страницу о хобби вязания с полным оформлением

#### **Глава 2: Стали художниками цвета**

- **Узнали:** 4 способа задавать цвет:
  - Простые слова: `red`, `blue`, `green`
  - HEX-коды: `#FF0000`, `#8B4513`
  - RGB: `rgb(255, 0, 0)`
  - RGBA с прозрачностью: `rgba(0, 0, 0, 0.5)`
- **Научились:** Различать `color` (цвет текста) и `background-color` (цвет фона)
- **Создали:** Поздравительные открытки с разными цветовыми схемами

#### **Глава 3: Стали мастерами текста**

- **Узнали:** Как выбирать и подключать красивые шрифты с Google Fonts
- **Научились:** Управлять:
  - Размером: `font-size: 16px;` (и `px`, `em`, `rem`, `%`)
  - Жирностью: `font-weight: 700;` (от 100 до 900)

- Наклоном: `font-style: italic;`
  - Выравниванием: `text-align: center;`
- **Создали:** «Шрифтовую книгу» с коллекцией красивых шрифтов
- 

## 🏠 Часть 2: Мы стали дизайнерами пространства

### Глава 4: Поняли магию блоков

- **Узнали:** Что такое блочные элементы и «коробочная модель»
- **Научились:** Создавать отступы:

- `margin` — расстояние между рамками (как между фотографиями на стене)
- `padding` — поле внутри рамки (как паспарту в фотоальбоме)

- **Создали:** Персональную визитку с правильными отступами

### Глава 5: Добавили изящные детали

- **Узнали:** Как создавать рамки и тени
- **Научились:** Использовать:

- `border: 3px solid red;` — рамки (сплошные, пунктирные, точечные)
- `box-shadow: 5px 5px 10px grey;` — тени для объёма
- `border-radius: 10px;` — скругление углов

- **Создали:** Красивую визитку с рамкой и лёгкой тенью

### Глава 6: Научились управлять размерами

- **Узнали:** Как задавать ширину и высоту элементов
- **Научились:** Использовать:

- `width: 300px;` — фиксированная ширина
- `width: 50%;` — относительная ширина
- `max-width: 100%;` — чтобы элементы «влезали» в любой экран

- **Создали:** Адаптивные блоки, которые подстраиваются под размер окна
- 

## ✨ Часть 3: Мы освоили магию превращений

### Глава 7: Стали волшебниками классов

- **Узнали:** Что такое CSS-классы и зачем они нужны

- **Научились:** Создавать и применять классы:

- В CSS: `.важный-текст { color: red; }`
- В HTML: `<p class="важный-текст">...</p>`

- **Создали:** Стили для «особо важных объявлений» и «изящных подписей»

## Глава 8: Создали навигационное меню

- **Узнали:** Как превратить скучный список ссылок в красивую панель

- **Научились:** Использовать:

- `list-style: none;` — убираем маркеры списка
- `display: flex;` — делаем элементы в строку
- `padding` и `background-color` — для красивого оформления

- **Создали:** Профессиональное горизонтальное меню для сайта

## Глава 9: Добавили интерактивности

- **Узнали:** Что такое псевдокласс `:hover`

- **Научились:** Создавать эффекты при наведении мыши:

- Изменение цвета кнопок
- Подчёркивание ссылок
- Увеличение элементов

- **Создали:** Меню, которое «оживает» при наведении курсора
- 

## Часть 4: Мы овладели финальными штрихами и большими возможностями

### Глава 10: Стали мастерами работы с изображениями

- **Узнали:**

- Как делать изображения адаптивными (`max-width: 100%;`)
- Как добавлять рамки и скруглять углы (`border-radius`)
- Как создавать красивые подписи к фотографиям

- **Создали:** Фотогалерею с современным оформлением

### Глава 11: Открыли секрет Flexbox

- **Узнали:** Простую аналогию: контейнер = ящик для инструментов, элементы = инструменты

- **Научились:** Волшебной команде `display: flex;`

### ● Освоили:

- flex-direction — как раскладывать (в ряд/столбик)
- justify-content — как распределять вдоль
- align-items — как выравнивать поперёк
- flex-wrap — как переносить на новые строки

● **Создали:** Автоматическую фотогалерею, которая сама выстраивается

### ✓ Глава 12: Научились делать сайты адаптивными

● **Узнали:** Что такое медиа-запросы @media

● **Поняли аналогию:** Разные инструкции для большого телевизора и маленького смартфона

### ● Научились:

- Менять размеры шрифтов на маленьких экранах
- Перестраивать расположение элементов (ряд → столбик)
- Адаптировать меню, галереи, контент

● **Создали:** Страницу, которая красиво выглядит на любом устройстве

---

## Ваши достижения в цифрах:

Навык	Количество освоенных свойств	Созданных проектов
Работа с цветом	10+ свойств	5+ проектов
Работа с текстом	15+ свойств	7+ проектов
Работа с блоками	20+ свойств	10+ проектов
Flexbox	8+ свойств	3+ проектов
Адаптивность	5+ медиа-запросов	4+ проектов
<b>Итого</b>	<b>58+ свойств CSS</b>	<b>29+ реальных проектов</b>

---

## 🎯 Что вы теперь можете создавать САМОСТОЯТЕЛЬНО:

### 1. Личные проекты:

- 🎉 Сайт-поздравление ко дню рождения внука
- 💐 Фотоальбом вашего сада или хобби

-  Личный блог или дневник
-  Страница для дачного кооператива
-  Семейная летопись с фотографиями

## 2. Практические проекты:

-  Визитка для вашего маленького бизнеса (продажа вязаных вещей, пирогов и т.д.)
-  Календарь событий для клуба по интересам
-  Онлайн-рецептурник с вашими фирменными блюдами
-  Галерея ваших творческих работ
-  Электронные открытки для друзей и родных

## 3. Полезные проекты:

-  Список дел с красивым оформлением
  -  Страница с полезными телефонами (врачи, службы)
  -  Погодная страница с виджетами
  -  Страница с любимыми цитатами и стихами
  -  Интерактивная карта ваших путешествий
- 

## Ваш инструментарий на конец обучения:

1. **Редактор кода:** Visual Studio Code (умеете им пользоваться)
  2. **Браузер с инструментами разработчика:** Умеете проверять и отлаживать код (F12)
  3. **Файловая структура:** Знаете, как организовать файлы проекта
  4. **Библиотеки ресурсов:** Знаете, где брать шрифты, иконки, изображения
  5. **Система контроля:** Умеете делать backup копии и экспериментировать безопасно
- 

## Ваши новые суперсилы:

### 1. Суперсила Творца:

Вы можете взять обычный текст и превратить его в произведение искусства. Чёрно-белая информация становится цветной, живой, выразительной.

## **2. Суперсила Решателя проблем:**

Если что-то «сломалось» или выглядит не так — вы знаете, как это исправить. Вы не паникуете, а включаете режим детектива.

## **3. Суперсила Ученика:**

Вы научились учиться сложным вещам. Теперь любую новую технологию вы сможете освоить по той же схеме: маленькими шагами, с практикой, без страха.

## **4. Суперсила Учителя:**

Вы можете объяснить сложные вещи простыми словами (как мы это делали с Flexbox и медиа-запросами). Можете помочь внукам, друзьям, соседям.

## **5. Суперсила Создателя:**

У вас есть самый ценный навык XXI века — вы можете создавать что-то из ничего. Несколько строк кода → красивая страница → радость для других людей.

---



## **Как изменилось ваше восприятие интернета:**

**БЫЛО:** Интернет — это что-то магическое, непонятное, созданное «умными дядями»

**СТАЛО:** Интернет — это набор текстовых файлов, которые вы можете читать и изменять

**БЫЛО:** Смотрю на красивый сайт → «Какая красота! Как они это сделали?»

**СТАЛО:** Смотрю на красивый сайт → «О, здесь использован flexbox! А тут интересная тень. А это медиа-запрос для мобильных. Классно!»

**БЫЛО:** Зависимость от помощи (нужен внук, чтобы «сделать красиво»)

**СТАЛО:** Полная самостоятельность (сам/а могу изменить цвет, шрифт, расположение)

---



## Ваш следующий шаг (если захотите):

Теперь вы стоите на распутье. Вы можете:

### Вариант 1: Остаться на достигнутом

- Поддерживать созданные страницы
- Создавать новые простые проекты для себя и близких
- Быть «локальным экспертом» в своём кругу общения

### Вариант 2: Углубиться в веб-дизайн

#### 1. Изучить больше CSS:

- CSS Grid (ещё более мощная система сеток)
- CSS Анимации (движущиеся элементы)
- Продвинутые эффекты (градиенты, фильтры, blend-режимы)

#### 2. Изучить основы JavaScript:

- Интерактивные элементы (слайдеры, попапы)
- Валидация форм
- Динамическое изменение контента

#### 3. Освоить инструменты:

- Figma или Adobe XD (для прототипирования)
- Git (для контроля версий)
- Основы SEO (чтобы ваш сайт находили в поиске)

### Вариант 3: Специализироваться

● **Веб-дизайнер для пенсионеров:** Создавать простые, понятные сайты для своих ровесников

● **Создатель семейных сайтов:** Помогать семьям создавать онлайн-летописи

● **Волонтёр:** Помогать некоммерческим организациям создавать простые сайты

---

## Самое главное, что вы получили:

1. **Уверенность:** Вы доказали себе, что можете освоить сложную технологию в любом возрасте
  2. **Радость творчества:** Процесс создания красоты из кода приносит настоящее удовольствие
  3. **Связь с миром:** Вы говорите на современном языке с молодыми поколениями
  4. **Практическую пользу:** У вас есть реальный, полезный навык
  5. **Перспективу:** Вы открыли дверь в мир, который будет только развиваться
- 

## Ваше домашнее задание на всю жизнь:

1. **Создайте хотя бы один реальный проект** — для себя, для семьи, для друзей
  2. **Делитесь знаниями** — помогите хотя бы одному человеку освоить основы
  3. **Экспериментируйте** — раз в месяц пробуйте что-то новое в CSS
  4. **Наслаждайтесь процессом** — помните, это хобби, а не работа
  5. **Гордитесь собой** — вы совершили маленькое чудо!
- 

**Дорогой друг, наш учебник подошёл к концу, но ваше путешествие только начинается.** Вы стоите на пороге мира, где можете создавать красоту, помогать другим, самовыражаться и быть современным человеком в цифровую эпоху.

**Вы теперь не просто пользователь. Вы — творец.**

И помните самое важное правило, с которого мы начинали: **«У нас всё безопасно. Здесь нельзя ничего сломать безвозвратно».** Смело экспериментируйте, пробуйте, ошибайтесь и создавайте!

**Спасибо, что прошли этот путь со мной. Вы — молодец! **

## ● Где искать вдохновение? (Советуем простые и красивые сайты).

Дорогой друг, теперь, когда вы освоили основы CSS, настало время наполнить свой творческий сосуд вдохновением! Иногда кажется: "Вот бы сделать что-то красивое... но не знаю, как это должно выглядеть". Не переживайте! Все дизайнеры, даже самые профессиональные, постоянно ищут вдохновение вокруг себя. Давайте я покажу вам лучшие места, где можно "подсмотреть" идеи для своих проектов.

---

### 1. Сайты, созданные специально для вдохновения

#### A. Dribbble ([dribbble.com](https://dribbble.com)) - "Птичка" для дизайнеров

**Что это:** Как художественная галерея, но для веб-дизайна

**Что искать:**

- Набираем в поиске: "**simple website**" (простой сайт)
- Или: "**landing page**" (посадочная страница)
- Или: "**portfolio**" (портфолио)

**Примеры красивых, но понятных работ:**

text

1. Поиските: "Minimal portfolio" - увидите, как просто можно представить свои работы
2. Поиските: "Gardening website" - вдохновение для сайта о саде
3. Поиските: "Blog design" - идеи для вашего дневника

**Совет:** Не смотрите на сложные работы с анимациями. Ищите то, что вы могли бы повторить с вашими знаниями!

---

## **B. Behance ( [behance.net](https://www.behance.net) ) - Галерея творческих проектов**

**Что это:** Как большая выставка дизайнерских работ

**Что делать:**

1. Зайдите в раздел "**Web Design**" (Веб-дизайн)
2. Используйте фильтры: "**Minimal**" (минимализм), "**Clean**" (чистый стиль)
3. Посмотрите проекты с тегом "**Personal**" (личные)

**Пример для поиска:** "Family website" или "Hobby blog"

---

## **C. Awwwards ( [awwwards.com](https://awwwards.com) ) - "Оскар" для сайтов**

**Внимание!** Здесь очень сложные, навороченные сайты. Но...

**Как использовать правильно:**

1. Нажмите в меню "**Sites**" (Сайты)
2. Выберите фильтр "**Minimal**" (Минимализм)
3. Или: "**Classic**" (Классика)

**Сайт дня часто бывает слишком сложным, но в категориях можно найти простые жемчужины!**

---



## 2. Реальные сайты, которые вам понравятся

### Сайты о природе и хобби (простые и красивые):

#### 1. Сады и цветы:

- **The Sill** ( [thesill.com](http://thesill.com) ) - сайт о комнатных растениях

Что подсмотреть: как красиво показывать фотографии растений

- **Gardenista** ( [gardenista.com](http://gardenista.com) ) - журнал о садоводстве

Что подсмотреть: тёплые цвета, уютное оформление

#### 2. Рукоделие и творчество:

- **Ravelry** ( [ravelry.com](http://ravelry.com) ) - сообщество вязальщиц

Что подсмотреть: как организовать галерею работ

- **Etsy** ( [etsy.com](http://etsy.com) ) - ручная работа

Что подсмотреть: карточки товаров, простые описания

#### 3. Кулинария и рецепты:

- **Smitten Kitchen** ( [smittenkitchen.com](http://smittenkitchen.com) ) - кулинарный блог

Что подсмотреть: как оформлять рецепты с фотографиями

- **Minimalist Baker** ( [minimalistbaker.com](http://minimalistbaker.com) )

Что подсмотреть: чистый дизайн, акцент на контенте

---

### Личные блоги и портфолио (то, что вы можете создать сами):

#### 4. Блоги "обычных" людей:

- **Search: "personal blog design simple"** в Google
- Посмотрите, как люди оформляют свои дневники
- Обратите внимание на сочетания цветов и шрифтов

#### 5. Сайты фотографов-любителей:

- Часто делаются очень просто: фото + краткое описание

- Идеально для создания семейного фотоальбома онлайн
- 

## **3. Специально для начинающих - сайты с ПОНЯТНЫМИ примерами**

### **A. CSS Design Awards ( [cssdesignawards.com](http://cssdesignawards.com) )**

Здесь есть категория "**Website of the Day**", но главное — смотрите не победителей, а **номинантов**. Они часто проще и понятнее.

**Как искать:** В фильтрах выберите "**Minimal**" и "**Blog**"

### **B. SiteInspire ( [siteinspire.com](http://siteinspire.com) )**

Лучший сайт для новичков! Есть удобные фильтры:

**Правильные фильтры для вас:**

- **Style:** Minimal (Минимализм), Playful (Игриный), Simple (Простой)
- **Type:** Blog (Блог), Portfolio (Портфолио), Personal (Личный)

### **C. Land-book ( [land-book.com](http://land-book.com) )**

Только посадочные страницы (landing pages), которые обычно делаются просто и понятно.

**Что подсмотреть:**

- Как организовать информацию
  - Какие цвета сочетаются
  - Как расположить текст и изображения
-

## 4. Что конкретно "подсматривать" на этих сайтах

Когда вы смотрите на красивый сайт, обращайте внимание не на всю сложность, а на отдельные элементы:

### Цветовые палитры:

text

Сайт о саде → Пастельные зелёные, бежевые, кремовые

Сайт о кулинарии → Тёплые коричневые, оранжевые, красные

Личный блог → Спокойные синие, серые, белые

**Задание:** Откройте понравившийся сайт, посмотрите на 3 основных цвета и запишите их коды (F12 → Elements → Styles).

### Шрифтовые пары:

Один шрифт для заголовков + один для текста. Частые сочетания:

- **Playfair Display** (заголовки) + **Open Sans** (текст)
- **Merriweather** (заголовки) + **Source Sans Pro** (текст)
- **Georgia** (заголовки) + **Arial** (текст) - самое простое!

### Расположение элементов:

- Как размещены фотографии относительно текста
- Какие отступы между разделами
- Как оформлены заголовки и подзаголовки

### Навигация:

- Как сделано меню (простое или сложное)
  - Как выделены активные пункты
  - Какие иконки используются
-

## 5. Практическое упражнение: "Охота за вдохновением"

### Шаг 1: Создайте "Блокнот вдохновения"

Создайте на компьютере папку "**Моё вдохновение**" и внутри:

1. **Файл "Цвета.txt"** - записывайте понравившиеся сочетания:

text

#8B4513 (коричневый) + #F5F5DC (бежевый) - с сайта о садоводстве

#2E8B57 (зелёный) + #FFFACD (светло-жёлтый) - с сайта о растениях

2. **Файл "Шрифты.txt"** - записывайте удачные пары:

text

Заголовок: "Playfair Display", Текст: "Open Sans" - с личного блога

Заголовок: "Georgia", Текст: "Arial" - с новостного сайта

3. **Папка "Скриншоты"** - делайте скриншоты (Print Screen) понравившихся элементов:

- Красивая шапка сайта
- Удачное расположение фото и текста
- Красивая кнопка
- Интересное меню

### Шаг 2: Анализ одного простого сайта

Возьмите например: [smittenkitchen.com](http://smittenkitchen.com) (кулинарный блог)

**Разберите его по частям:**

text

1. Шапка:

- Какой цвет фона? (#ffffff - белый)
- Какой шрифт в заголовке? ("Times New Roman")
- Какие цвета в меню? (#333333 - тёмно-серый)

2. Основной контент:

- Как расположены рецепты? (Сетка)
- Какие отступы между карточками? (20px)

- Как оформлены фотографии? (Скруглённые углы 8px)

### 3. Подвал:

- Какая информация там есть? (Автор, год, ссылки)
- Какой цвет фона? (#f8f8f8 - светло-серый)

## Шаг 3: Создайте "референс-доску"

В программе Paint или даже в Word создайте коллаж из того, что вам нравится:

- Скриншоты красивых элементов
  - Выписки цветовых кодов
  - Примеры шрифтов
  - Схемы расположения элементов
- 

## ★ 6. Специально для вас: 10 сайтов с ИДЕАЛЬНОЙ простотой

Вот список сайтов, которые одновременно красивы и достаточно просты, чтобы вы могли понять, как они сделаны:

1. **Блог о чтении:** [zenhabits.net](http://zenhabits.net) (максимально просто!)
  2. **Фотограф-натуралист:** [marcuswestberg.com](http://marcuswestberg.com) (просто фото + минимум текста)
  3. **Садовый блог:** [growingwithplants.com](http://growingwithplants.com) (милый и простой)
  4. **Личный дневник:** [austinkleon.com](http://austinkleon.com) (как тетрадка)
  5. **Семейные рецепты:** [thepioneerwoman.com](http://thepioneerwoman.com) (уютно и понятно)
  6. **Художник-акварелист:** [teelaunch.com/blog](http://teelaunch.com/blog) (ярко и просто)
  7. **Путешественник на пенсии:** [seniornomads.com](http://seniornomads.com) (истории + фото)
  8. **Вязание:** [purlsoho.com](http://purlsoho.com) (мягкие цвета, понятная структура)
  9. **Дачный блог:** [gardenbetty.com](http://gardenbetty.com) (много фото, минимум "наворотов")
  10. **Личная страничка:** [bradfrost.com](http://bradfrost.com) (чисто, понятно, современно)
-



## 7. Как "заглянуть под капот" понравившегося сайта

### Волшебная кнопка F12!

Когда вам нравится какой-то элемент на сайте:

1. **Нажмите F12** (или Ctrl+Shift+I)
2. **Нажмите значок лупы** (Select element)
3. **Кликните на понравившийся элемент** на странице
4. **Справа увидите CSS-стили** этого элемента!

**Что можно узнать:**

- Какие цвета используются (color, background-color)
- Какие шрифты (font-family)
- Какие отступы (margin, padding)
- Какие размеры (width, height)

**Пример:**

Понравилась красивая кнопка на сайте:

text

1. F12 → лупа → клик на кнопку

2. Видим справа:

```
background-color: #4CAF50;  
color: white;  
padding: 12px 24px;  
border-radius: 8px;  
font-family: 'Arial', sans-serif;
```

3. Копируем эти стили в свой проект!

---



## 8. Где НЕ искать вдохновение (пока)

**Избегайте на начальном этапе:**

1. **Сайты крупных компаний** (Apple, Microsoft) - слишком сложно
2. **Интернет-магазины** (Amazon, Ozon) - слишком много элементов

3. **Социальные сети** (Facebook, VK) - слишком специфично
4. **Сайты с сложной анимацией** - это следующий уровень

## Почему?

Эти сайты создают целые команды профессионалов, используют сложные технологии (JavaScript, фреймворки), которые вы ещё не изучали. Вы можете запутаться и подумать: "Я никогда так не смогу!"

**Помните:** Вы учитесь ходить, а они уже бегают марафоны. Начните с простой ходьбы!

---



## 9. Вдохновение в оффлайн-мире

Вдохновение для сайта можно найти не только в интернете!

### Журналы и газеты:

- **Дачные журналы** - вёрстка, сочетания цветов
- **Журналы по рукоделию** - как показать работы
- **Поздравительные открытки** - идеи для небольших сайтов-поздравлений

### Природа вокруг:

- **Ваш сад** - цветовые палитры (зелёный+коричневый+цветы)
- **Времена года** - осенние цвета, весенние пастельные тона
- **Любимые места** - фотографии для фона сайта

### Повседневная жизнь:

- **Альбомы с фотографиями** - как организовать галерею
  - **Кулинарные книги** - как структурировать рецепты
  - **Семейные реликвии** - идеи для "семейного музея" онлайн
-



## 10. Практическое задание: "Неделя вдохновения"

### День 1: Цвета

- Найдите 3 сайта с красивыми цветовыми схемами
- Запишите по 2 основных цвета с каждого
- Создайте в CSS файле палитру этих цветов

### День 2: Шрифты

- Найдите 3 сайта с красивыми, читаемыми шрифтами
- Определите, какие шрифты используются (F12)
- Создайте HTML-страницу с этими шрифтовыми параметрами

### День 3: Фотографии

- Найдите 3 сайта, где красиво представлены фотографии
- Проанализируйте: размер, рамки, расположение, отступы
- Создайте простую галерею по мотивам увиденного

### День 4: Навигация

- Найдите 3 простых и понятных меню
- Проанализируйте: цвета, шрифты, hover-эффекты
- Создайте своё меню, вдохновлённое увиденным

### День 5: Структура

- Найдите сайт с простой и понятной структурой
- Нарисуйте на бумаге схему: что где расположено
- Создайте HTML-каркас по этой схеме

### День 6: Детали

- Найдите красивые детали: кнопки, разделители, цитаты
- Создайте коллекцию этих элементов в отдельном CSS-файле

## День 7: Свой проект

- Соберите всё лучшее из недели
  - Создайте страницу "Мои идеи для сайта"
  - Оформите её с использованием найденного вдохновения
- 

### ❤️ Важное напоминание:

1. **Не копируйте слепо** - берите идеи, а не код
2. **Адаптируйте под себя** - то, что хорошо для одного сайта, может не подойти для вашего
3. **Начинайте с простого** - не пытайтесь повторить сложные элементы
4. **Экспериментируйте** - возьмите идею и попробуйте сделать её по-своему
5. **Наслаждайтесь процессом** - поиск вдохновения должен быть удовольствием, а не работой

**Дорогой друг, теперь у вас есть целая карта мест, где можно найти вдохновение!** Помните: все великие художники изучали работы мастеров до них, все дизайнеры смотрят на работы коллег. Это нормально, полезно и правильно.

**Ваша задача сейчас:** не создать шедевр с первого раза, а научиться видеть красоту, понимать, как она создана, и пробовать делать что-то похожее своими руками.

## ● Куда двигаться дальше? (Добрые рекомендации для следующего шага).

Дорогой друг, вы закончили наш учебник — это огромное достижение! Но как в любом интересном путешествии, финиш одной тропы — это только начало новых дорог. Давайте вместе посмотрим, куда вы можете отправиться дальше. Я подготовил для вас несколько маршрутов разной сложности — от прогулочных тропинок до настоящих горных маршрутов. Выбирайте то, что вам по душе!

---

## Маршрут 1: "Прогулка в парке" (Для тех, кто хочет закрепить и наслаждаться)

**Если вы думаете: "Я хочу использовать то, что уже знаю, не усложняй"**

**Этот маршрут для вас!**

**Что делать:**

### 1. Создайте "Портфолио простых проектов"

Сделайте 5-7 реальных страничек:

-  **Личный дневник** о ваших днях (каждый день новая запись с фото)
-  **Сезонный календарь садовода** (что сажать, когда поливать)
-  **Семейная летопись** (фото, истории, даты)
-  **Каталог ваших работ** (вязание, вышивка, поделки)
-  **Кулинарная книга** с вашими рецептами
-  **Фотоальбом года** (12 месяцев, 12 страниц)
-  **Сайт-поздравление** к ближайшему празднику

**Цель:** Не учить новое, а довести до автоматизма уже изученное.

## **2. Станьте "CSS-гуру" в своём кругу**

- Помогите другу сделать визитку
- Создайте страничку для дачного кооператива
- Сделайте "меню дня" для семейных обедов
- Оформите фотоальбом для внуков

## **3. Участвуйте в "Неделях творчества"**

Каждую неделю делайте один маленький проект:

- Неделя красного (всё в красных тонах)
- Неделя закруглённых углов
- Неделя теней и объёма
- Неделя красивых шрифтов

### **Ресурсы для этого маршрута:**

 **Ваш собственный учебник** — перечитывайте сложные моменты

 [Coolors.co](#) — генератор цветовых палитр

 [Fontpair.co](#) — подбор сочетаний шрифтов

---



## **Маршрут 2: "Поход с гидом" (Плавное углубление)**

**Если вы думаете: "Хочу знать чуть больше, но без стресса"**

**Идеальный выбор!**

### **Что изучать дальше:**

#### **1. CSS Grid — "старший брат" Flexbox**

**Зачем:** Если Flexbox — это умный ящик для инструментов, то CSS Grid — целый организованный склад!

## Что это даст:

- Создавать сложные макеты проще
- Располагать элементы точно по сетке
- Делать настоящие журнальные вёрстки

**Простая аналогия:** Flexbox раскладывает инструменты в ящике, а Grid расставляет мебель в комнате по разметке на полу.

## С чего начать:

css

```
/* Простейший Grid */  
.container {  
    display: grid;  
    grid-template-columns: 1fr 1fr 1fr; /* 3 колонки равной ширины */  
    gap: 20px; /* Промежутки */  
}
```

## Ресурсы:

- 🎮 **CSS Grid Garden** ([cssgridgarden.com](https://cssgridgarden.com)) — игра-обучение
- 📖 "Изучаем CSS Grid" от MDN (на русском)

## 2. Псевдоклассы и псевдоэлементы

Вы уже знаете `:hover`. Пора узнать его "родственников"!

## Что изучать:

- `:nth-child()` — выбрать каждый 3-й элемент
- `::before` и `::after` — добавить content до/после элемента
- `:focus` — стили для поля ввода при клике
- `:checked` — для отмеченных чекбоксов

## Пример применения:

css

```
/* Красивые списки */  
li:nth-child(odd) { background: #f0f0f0; } /* Нечётные строки */  
li:nth-child(even) { background: white; } /* Чётные строки */
```

```
/* Добавить иконку перед пунктами */
li::before {
  content: "✓ ";
  color: green;
}
```

### 3. CSS-переменные (Custom Properties)

**Зачем:** Чтобы не искать по всему коду, где вы использовали цвет #8B4513.

**Как работает:**

css

```
/* Объявляем переменные */
:root {
  --main-color: #8B4513;
  --secondary-color: #D2B48C;
  --font-size: 18px;
}

/* Используем */
h1 {
  color: var(--main-color);
  font-size: var(--font-size);
}
```

**Польза:** Изменили значение в одном месте — поменялось везде!

---



## Маршрут 3: "Восхождение на вершину" (Для самых смелых)

**Если вы думаете: "Хочу создать по-настоящему крутые сайты!"**

**Добро пожаловать в мир продвинутого веб-дизайна!**

# Этап 1: Продвинутый CSS

## 1. CSS-анимации и переходы

**Что это:** Заставить элементы двигаться плавно!

**Простой пример — "оживающая" кнопка:**

css

```
.button {  
    background: #3498db;  
    transition: all 0.3s ease; /* Плавное изменение ВСЕХ свойств */  
}  
  
.button:hover {  
    background: #2980b9;  
    transform: scale(1.05); /* Увеличивается на 5% */  
    box-shadow: 0 5px 15px rgba(0,0,0,0.3);  
}
```

**Что ещё можно анимировать:**

- Появление элементов (fade in)
- Слайд-шоу фотографий
- Подсказки, которые выезжают
- Загрузочные индикаторы

## 2. Flexbox и Grid — продвинутый уровень

- Вложенные Grid-сетки
- Автоматическое размещение (auto-fit, auto-fill)
- Выравнивание по областям (grid-template-areas)

## 3. Современные CSS-эффекты

- **Градиенты:** background: linear-gradient(45deg, red, blue);
- **Тени текста:** text-shadow: 2px 2px 4px rgba(0,0,0,0.5);
- **Фильтры для изображений:** filter: blur(5px) brightness(1.2);
- **Обтекание текста:** shape-outside: circle();

## Этап 2: Основы JavaScript (без паники!)

**Важно:** JavaScript — это НЕ обязательно программирование! Это способ сделать сайт интерактивным.

**Что можно сделать с базовым JS:**

1. **Слайдер фотографий** (листать кнопками "вперёд/назад")
2. **Модальные окна** (всплывающие окошки)
3. **Аккордеон** (разворачивающиеся блоки)
4. **Валидация форм** (проверка email, телефона)
5. **Тёмная/светлая тема** (переключатель)

**Пример: Кнопка "Показать ещё"**

javascript

```
// HTML: <button onclick="showMore()">Показать ещё</button>
// HTML: <div id="hiddenText" style="display:none">Скрытый текст</div>

function showMore() {
  const hiddenText = document.getElementById('hiddenText');
  hiddenText.style.display = 'block';
  alert('Текст показан!');
}
```

**Ресурсы для изучения JavaScript:**

- 📚 "JavaScript для детей" — несмотря на название, идеально для начинающих
- 🎮 Codecademy (бесплатный курс JS)
- 🎥 Канал "Владилена Минина" на YouTube (объясняет очень понятно)

## Этап 3: Инструменты профессионалов

### 1. Система контроля версий Git

**Зачем:** Чтобы не бояться "сломать" работающий сайт.

**Аналогия:** Git — это как "сохранения" в компьютерной игре. Можно вернуться к любому моменту.

## **Базовые команды:**

```
bash  
  
git add .      # "Подготовить" изменения  
git commit -m "Добавил галерею" # "Сохранить" с комментарием  
git push      # Загрузить на GitHub
```

**Сайт для хранения кода:** [GitHub.com](#) (бесплатно)

## **2. Фреймворки и библиотеки**

- **Bootstrap** — готовые компоненты (кнопки, меню, формы)
- **jQuery** — упрощённый JavaScript (для начинающих)
- **React/Vue** — для сложных интерфейсов (это уже серьёзно!)

**Совет:** Начните с Bootstrap. Он использует знакомые вам классы CSS.

---

## **Маршрут 4: "Специализация" (Найти свою нишу)**

**Выберите направление, которое вам ближе:**

**Вариант А: Дизайнер для "серебряного" возраста**

**Идея:** Создавать сайты для людей вашего поколения.

**Что изучать:**

- Принципы доступности (accessibility)
- Крупные шрифты, контрастные цвета
- Простые, интуитивные интерфейсы
- Помощь с цифровизацией для пенсионеров

**Проекты:**

- Сайты для клубов по интересам
- Онлайн-дневники для бабушек и дедушек

- Фотоархивы для семей

## **Вариант Б: Создатель "семейных цифровых архивов"**

**Идея:** Помогать семьям сохранять историю.

### **Что изучать:**

- Организация фотоархивов
- Создание генеалогических деревьев
- Оцифровка старых фотографий
- Публикация мемуаров

### **Проекты:**

- Семейные сайты-летописи
- Онлайн-альбомы для нескольких поколений
- Страницы памяти

## **Вариант В: "Хобби-эксперт"**

**Идея:** Стать лучшим в создании сайтов для вашего хобби.

### **Например, если вы садовод:**

- Изучить, как лучше показывать растения
  - Создать базу данных сортов
  - Сделать интерактивный календарь посадок
  - Разработать шаблоны для других садоводов
- 

## **Карта ресурсов для продолжения обучения**

### **Бесплатные курсы (на русском):**

#### **1. HTML Academy ( [htmlacademy.ru](http://htmlacademy.ru) )**

- Интерактивные тренажёры

- Много практики
- От простого к сложному

## 2. Stepik ( [stepik.org](https://stepik.org) )

- Курс "Веб-разработка для начинающих"
- Бесплатные сертификаты
- Сообщество студентов

## 3. Hexlet ( [hexlet.io](https://hexlet.io) )

- Бесплатные вводные курсы
- Профессиональный подход
- Поддержка сообщества

## YouTube-каналы:

### Для начинающих:

-  "Веб-разработка от А до Я" (Владилен Минин)
-  "Фрилансер по жизни" (IT-фриланс простыми словами)
-  "Гоша Дударь" (очень подробные уроки)

### Про CSS:

-  "Код.Дизайн" (красивые CSS-эффекты)
-  "WebDev" (практические приёмы)

## Книги (можно найти в библиотеках):

### Начальный уровень:

-  "HTML и CSS. Разработка и дизайн веб-сайтов" (Джон Дакетт)
-  "Изучаем HTML, XHTML и CSS" (Эрик Фримен)

### Средний уровень:

-  "CSS. Карманный справочник" (Эрик Мейер)
  -  "Секреты CSS" (Лия Вери)
-



## Проекты для портфолио (от простого к сложному)

### Уровень 1: "Я это могу!"

1. **Визитка** (имя, фото, контакты, хобби)
2. **Рецепт** с пошаговыми фото
3. **Фотогалерея** одного события
4. **Список дел** на неделю
5. **Поздравительная открытка**

### Уровень 2: "Это уже серьёзно!"

1. **Личный блог** с 5-7 статьями
2. **Сайт-портфолио** ваших работ
3. **Календарь событий** на месяц
4. **Интерактивная карта** ваших путешествий
5. **Сайт для клуба** по интересам

### Уровень 3: "Я — веб-дизайнер!"

1. **Многостраничный сайт** с меню
  2. **Адаптивный сайт** (от телефона до компьютера)
  3. **Сайт с формой обратной связи**
  4. **Интернет-магазин** (каталог + корзина)
  5. **Веб-приложение** (например, планировщик сада)
- 



## Сообщества для поддержки

### Где общаться с такими же учениками:

#### Русскоязычные:

- **Хабр Q&A** ([qna.habr.com](https://qna.habr.com)) — задавайте вопросы
- **Клуб веб-разработчиков** в вашем городе (ищите в VK)

- **Telegram-чаты** по веб-разработке

### **Международные:**

- **freeCodeCamp Forum** (форум на английском, но очень добрый)
- **CodeNewbie** (сообщество для начинающих)
- **Dev.to** (блоги разработчиков, можно читать через переводчик)

### **Как правильно задавать вопросы:**

1. **Покажите код**, который не работает
  2. **Объясните**, что должно было получиться
  3. **Расскажите**, что уже пробовали сделать
  4. **Будьте вежливы** — все когда-то начинали
- 



### **План на первые 3 месяца после учебника**

#### **Месяц 1: "Закрепление основ"**

- **Неделя 1-2:** Повторить Flexbox, сделать 3 проекта
- **Неделя 3-4:** Освоить медиа-запросы, сделать адаптивный сайт

#### **Месяц 2: "Расширение горизонтов"**

- **Неделя 1-2:** Изучить CSS Grid (основы)
- **Неделя 3-4:** Добавить простые анимации к проектам

#### **Месяц 3: "Первый серьёзный проект"**

- Создать полноценный сайт о своём хобби
  - Сделать его адаптивным
  - Добавить интерактивные элементы
  - Выложить в интернет (бесплатный хостинг: GitHub Pages)
-

## **Как превратить хобби в маленький бизнес (если захотите)**

### **Услуги, которые вы можете предлагать:**

#### **Для друзей и знакомых:**

1. **"Оцифровка семейного архива"** — 3000-5000 руб.
2. **"Сайт-визитка для пенсионера"** — 2000-4000 руб.
3. **"Онлайн-альбом для событий"** — 1500-3000 руб.

#### **Для местных сообществ:**

1. **Сайт дачного кооператива** — 5000-10000 руб.
2. **Страница для клуба по интересам** — 3000-7000 руб.
3. **Календарь событий для ТСЖ** — 4000-8000 руб.

### **Как находить заказы:**

1. **Расскажите о своём умении** на собраниях, в поликлинике, в магазине
  2. **Покажите примеры работ** на телефоне
  3. **Предложите сделать первый проект недорого** для портфолио
  4. **Попросите рекомендации** у довольных клиентов
- 

## **Самое главное — сохранить мотивацию**

### **Что делать, если опускаются руки:**

1. **Вспомните свой первый проект** — как вы радовались, когда получилось!
2. **Посмотрите на работы недельной давности** — вы уже стали лучше!
3. **Сделайте перерыв** — иногда нужно отдохнуть 2-3 дня
4. **Найдите "брата по разуму"** — учитесь вместе с кем-то
5. **Поставьте маленькую цель** — не "выучить CSS", а "сделать красивую кнопку"

## **Правило 20 минут:**

Каждый день уделяйте веб-дизайну хотя бы 20 минут. Можно:

- Посмотреть урок на YouTube
- Почитать статью
- Сделать маленький элемент (кнопку, карточку)
- Поэкспериментировать с цветами

**20 минут × 30 дней = 10 часов в месяц!** За год это 120 часов — целых 5 дней обучения!

---

## **Ваш план "Дальше" на выбор:**

### **Если вам больше всего понравилось:**

- **Работа с цветами** → изучайте Color Theory, психологию цвета
- **Вёрстка текста** → изучайте типографику, леттеринг
- **Создание макетов** → изучайте UI/UX дизайн
- **Работа с фото** → изучайте основы фотообработки
- **Программирование** → начинайте с JavaScript

### **Если вы хотите конкретный результат:**

- **"Хочу сайт о саде"** → изучайте, как лучше показывать растения
  - **"Хочу семейный архив"** → изучайте организацию медиа-контента
  - **"Хочу помогать другим"** → изучайте доступность (accessibility)
  - **"Хочу зарабатывать"** → изучайте фриланс, основы бизнеса
- 

## **Финальные напутствия:**

1. **Вы уже победитель** — большинство людей даже не начинают
2. **Учиться можно в любом возрасте** — ваш опыт только помогает
3. **Ошибки — это нормально** — без них не бывает прогресса

4. **Делитесь знаниями** — обучая других, вы учитеь сами
5. **Наслаждайтесь процессом** — это должно приносить радость!

**Дорогой друг, ваше путешествие в мир веб-дизайна только начинается.** Вы стоите у подножия огромной, прекрасной горы под названием "Веб-разработка". Не нужно взбираться на вершину за один день. Поднимайтесь понемногу, наслаждайтесь видами, делайте привалы, когда нужно.

**Самое главное — вы сделали первый и самый сложный шаг: начали.**

Теперь у вас есть карта, компас и базовые навыки. Куда идти дальше — решать только вам. Но куда бы вы ни пошли, помните: вы можете это сделать!

**Удачи вам, вдохновения и много красивых сайтов на вашем творческом пути!** 

## **Приложения (полезные шпаргалки):**

### **1. Мини-словарик: Перевод самых частых терминов с «компьютерного» на русский.**

Дорогой друг, компьютерный мир полон странных слов, которые могут пугать. Но на самом деле за каждым из них стоит простое понятие! Давайте переведём эти "загадочные" термины на нормальный человеческий язык.

---



### **Основные понятия — как азбука**

#### **A. Про компьютер и интернет**

<b>Компьютерное слово</b>	<b>Простой перевод</b>	<b>Пример из жизни</b>
<b>Браузер</b>	<b>Окно в интернет</b>	Программа, в которой вы смотрите сайты (Chrome, Яндекс.Браузер, Firefox)
<b>Сайт</b>	<b>Книга/журнал в интернете</b>	Страница в интернете, как газета, но на экране
<b>Страница</b>	<b>Лист в книге сайта</b>	Один "листик" сайта, который вы видите на экране
<b>Интернет</b>	<b>Всемирная паутина/почта</b>	Как телефонная сеть, но для компьютеров
<b>Сервер</b>	<b>Библиотекарь интернета</b>	Компьютер, который хранит сайты и выдаёт их по запросу
<b>Хостинг</b>	<b>Аренда места в интернете</b>	Как аренда земли для строительства дома, но для сайта
<b>Домен</b>	<b>Адрес сайта</b>	Как домашний адрес, но для сайта (например, <a href="http://moy-sad.ru">moy-sad.ru</a> )

## Б. Про файлы и папки

Компьютерное слово	Простой перевод	Пример из жизни
Файл	Документ/записка	Как лист бумаги с информацией, но в компьютере
Папка	Папка для документов	Как картонная папка на столе, где лежат файлы
Расширение	Тип документа	Как подпись на конверте (.html — сайт, .jpg — фото)
Рабочий стол	Письменный стол	Главный экран, где лежат ярлыки нужных программ
Ярлык	Записка "где лежит"	Как бумажка с адресом, а не сам предмет



## Словарь веб-дизайнера (то, что вы уже знаете!)

## А. HTML — "Скелет" сайта

Компьютерное слово	Простой перевод	Простыми словами
HTML	Каркас/скелет сайта	Кирпичи и балки дома (что где находится)
Тег	Ярлычок/бирка	Табличка на двери комнаты: "кухня", "спальня"
<h1> - <h6>	Заголовки разного уровня	Как заголовки в газете: главный, поменьше, ещё меньше
<p>	Абзац	Обычный текст, как в письме или книге
<div>	Коробка/ящик	Картонная коробка, в которую можно положить что угодно
<a>	Ссылка/дверь	Дверь в другую комнату (страницу)

Компьютерное слово	Простой перевод	Простыми словами
<img>	Фотография/картинка	Рамка с фото на стене
<ul>/<li>	Список	Список покупок или дел
Атрибут	Характеристика/свойство	Как цвет двери или материал стола

## Б. CSS — "Одежда и мебель" сайта

Компьютерное слово	Простой перевод	Аналогия из жизни
CSS	Дизайн/оформление	Обои, краска, мебель в доме
Свойство	Команда браузеру	Как сказать: "Покрась эту стену в синий!"
Значение	Как именно сделать	"В синий" (а не в красный или зелёный)
Селектор	Указатель/адресат	Кому даём команду ("всем стульям", "только этому столу")
Класс	Имя группы	Как назвать группу предметов: "красные стулья"
ID	Личный номер/имя	Как имя человека (的独特的)
Правило	Инструкция	Полная фраза: "Все красные стулья — поставь в угол"

## В. Конкретные CSS-свойства

Компьютерное слово	Простой перевод	Наглядный пример
color	Цвет чернил	Какими чернилами писать текст
background-color	Цвет бумаги	На какой бумаге писать (белая, цветная)
font-family	Почерк/шрифт	Какой почерк использовать (красивый,

## Компьютерное слово

## Простой перевод Наглядный пример

строгий)

<b>font-size</b>	<b>Размер букв</b>	Крупно или мелко писать
<b>text-align</b>	<b>Выравнивание</b>	Писать по центру, слева или справа
<b>margin</b>	<b>Отступ снаружи</b>	Расстояние между картинами на стене
<b>padding</b>	<b>Отступ внутри</b>	Поля в рамке фотографии (от фото до рамки)
<b>border</b>	<b>Рамка</b>	Рамка вокруг картины или фотографии
<b>border-radius</b>	<b>Закругление углов</b>	Скруглить углы рамки (современный вид)
<b>display: flex</b>	<b>Умная расстановка</b>	Специальная полка, которая сама расставляет предметы ровно
<b>@media</b>	<b>Условие "если"</b>	Инструкция: "Если экран маленький — сделай вот так"



## Технические термины, которые встречаются

### A. При работе с кодом

## Компьютерное слово

## Простой перевод

## Объяснение

<b>Код</b>	<b>Рецепт/инструкция</b>	Пошаговое описание, как сделать сайт
<b>Синтаксис</b>	<b>Грамматика/правила</b>	Как правильно писать команды (где точка, где запятая)
<b>Комментарий</b>	<b>Заметка на полях</b>	Запись для себя, которую компьютер игнорирует
<b>Ошибка (баг)</b>	<b>Опечатка/недосмотр</b>	Как опечатка в рецепте — торт не получается
<b>Отладка</b>	<b>Исправление ошибок</b>	Поиск и исправление опечаток в рецепте

<b>Компьютерное слово</b>	<b>Простой перевод</b>	<b>Объяснение</b>
<b>Валидация</b>	<b>Проверка правильности</b>	Проверить, всё ли по правилам написано

## Б. При просмотре сайтов

<b>Компьютерное слово</b>	<b>Простой перевод</b>	<b>Что это значит</b>
<b>Адаптивность</b>	<b>Приспособляемость</b>	Сайт подстраивается под размер экрана (телефон/компьютер)
<b>Интерфейс</b>	<b>Средство общения</b>	Всё, с чем вы взаимодействуете: кнопки, меню, формы
<b>Навигация</b>	<b>Путеводитель</b>	Меню и ссылки, которые помогают перемещаться по сайту
<b>Хлебные крошки</b>	<b>Следы пути</b>	Строка типа "Главная → Сад → Розы" — где вы находитесь
<b>Футер</b>	<b>Подвал сайта</b>	Нижняя часть страницы (контакты, авторские права)
<b>Хедер</b>	<b>Шапка сайта</b>	Верхняя часть страницы (логотип, главное меню)
<b>Попап</b>	<b>Всплывающее окно</b>	Окошко поверх страницы (например, подписка на новости)



## Слова, которые пугают, но просты

### Самые страшные (на первый взгляд) термины:

<b>Пугающее слово</b>	<b>На самом деле это...</b>	<b>Простое объяснение</b>
<b>Алгоритм</b>	<b>Последовательность</b>	Рецепт приготовления борща (делай

<b>Пугающее слово</b>	<b>На самом деле это...</b>	<b>Простое объяснение</b>
	<b>действий</b>	шаг 1, потом шаг 2...)
<b>Фреймворк</b>	<b>Набор готовых инструментов</b>	Готовый набор мебели "собери сам" (IKEA для сайтов)
<b>Библиотека</b>	<b>Коллекция готовых решений</b>	Как книга с готовыми шаблонами открыток
<b>Функция</b>	<b>Маленькая программа-помощник</b>	Как кухонный комбайн: положил ингредиенты — получил результат
<b>Переменная</b>	<b>Ящик для хранения</b>	Как коробка с надписью "Сахар" — положил туда сахар, потом используешь
<b>Массив</b>	<b>Полка с ящиками</b>	Несколько коробок подряд: "Сахар", "Мука", "Соль"
<b>Цикл</b>	<b>Повторение</b>	Как повторять "помыть тарелку", пока тарелки не кончатся

## **Термины, которые звучат сложно, но вы их УЖЕ используете:**

<b>Сложное слово</b>	<b>Вы это уже делали, когда...</b>	<b>Пример из учебника</b>
<b>Декларативный язык</b>	Писали CSS-правила	color: red; (объявили, что цвет красный)
<b>Каскадность</b>	Видели, что последнее правило важнее	Если задали цвет в двух местах — работает последнее
<b>Наследование</b>	Элементы брали стили у родителей	Все <code>&lt;p&gt;</code> внутри <code>&lt;div&gt;</code> получили его стили
<b>Специфичность</b>	Класс важнее тега	. <code> важный</code> перебивает стили обычного <code>&lt;p&gt;</code>
<b>Псевдокласс</b>	Делали эффект при наведении	<code>:hover</code> — когда мышка наводится на элемент



## "Переводчик" для чтения уроков в интернете

### Когда читаете английские уроки:

Английское слово	Русский перевод	Пример использования
<b>container</b>	контейнер, контейнер	.container — основной блок страницы
<b>wrapper</b>	обёртка	.wrapper — то же, что контейнер
<b>header</b>	шапка	Верхняя часть сайта
<b>footer</b>	подвал	Нижняя часть сайта
<b>sidebar</b>	боковая панель	Боковая колонка
<b>navbar</b>	навигационная панель	Меню сайта
<b>button</b>	кнопка	Кликабельная кнопка
<b>link</b>	ссылка	Ссылка на другую страницу
<b>image</b>	изображение	Картинка, фото
<b>background</b>	фон	Фон элемента
<b>border</b>	рамка	Граница вокруг элемента
<b>shadow</b>	тень	Тень элемента
<b>animation</b>	анимация	Движение элемента
<b>responsive</b>	адаптивный	Подстраивающийся под экран

### Частые сочетания в названиях классов:

Английская часть	Перевод	Что означает
<b>-container</b>	-контейнер	Основной блок
<b>-wrapper</b>	-обёртка	Блок-обёртка
<b>-inner</b>	внутренний	Внутренняя часть
<b>-outer</b>	внешний	Внешняя часть
<b>-header</b>	-шапка	Верхняя часть блока

<b>Английская часть</b>	<b>Перевод</b>	<b>Что означает</b>
<b>-footer</b>	-подвал	Нижняя часть блока
<b>-content</b>	содержимое	Основное содержимое
<b>-menu</b>	меню	Меню, список
<b>-item</b>	элемент, пункт	Один элемент списка
<b>-active</b>	активный	Текущий, выбранный элемент
<b>-disabled</b>	отключённый	Неактивный элемент

## 🎮 Игровые термины для понимания процесса

### Программирование как игра:

<b>Игровой термин</b>	<b>Программистский аналог</b>	<b>Объяснение</b>
<b>Квест</b>	Задача/проект	Что нужно сделать (например, создать сайт)
<b>Инвентарь</b>	Набор знаний/навыков	То, что вы уже умеете (HTML, CSS, Flexbox)
<b>Прокачка навыков</b>	Обучение	Изучение нового (например, JavaScript)
<b>Босс</b>	Сложная задача	Что-то, что не получается с первого раза
<b>Чит-коды</b>	Готовые решения	Код с Stack Overflow или из примеров
<b>Сохранение</b>	Коммит в Git	Сохранение прогресса, чтобы можно было вернуться
<b>Новый уровень</b>	Новый проект	Когда переходите к более сложным задачам
<b>Рандом</b>	Случайность/баг	Когда что-то работает не так, как ожидалось



## Домашняя аналогия для сложных понятий

### Вся веб-разработка как строительство дома:

Веб-разработка Строительство дома Пояснение

HTML	Фундамент и стены	Что где находится (комнаты, окна, двери)
CSS	Отделка и мебель	Как всё выглядит (обои, краска, мебель)
JavaScript	Коммуникации и умный дом	Что может двигаться/меняться (свет, шторы, лифт)
Backend	Инженерные системы	То, что скрыто (проводка, трубы, отопление)
База данных	Архив документов	Где хранится информация (чертежи, счета, договоры)
Фреймворк	Типовой проект	Готовый проект дома, который можно адаптировать
Библиотека	Каталог мебели	Коллекция готовых элементов (двери, окна, сантехника)
API	Почтовый ящик	Способ общения с внешним миром (получить письмо, отправить)



### Краткий справочник "На каждый день"

### Когда нужно быстро вспомнить:

Если вы  
видите...      Это значит...      Что делать?

404	Страница не найдена	Проверить ссылку или вернуться назад
Ctrl+S	Сохранить	Всегда нажимать после изменений!
F5	Обновить страницу	Посмотреть изменения после сохранения
F12	Инструменты	Посмотреть код, найти ошибки

<b>Если вы видите...</b>	<b>Это значит...</b>	<b>Что делать?</b>
	разработчика	
<!-- ... -->	Комментарий в HTML	Компьютер это игнорирует, это заметка для себя
/* ... */	Комментарий в CSS	То же самое, но для стилей
#имя	ID элемента	Уникальное имя, как паспорт
.имя	Класс элемента	Имя группы, как "студенты группы А"
:hover	При наведении мыши	Стили, которые работают, когда курсор над элементом
@media	Условие	Стили только при определённых условиях (например, на телефоне)



## Как пользоваться этим словарём:

1. **Держите открытым** во время работы — как кулинарную книгу
2. **Добавляйте свои термины** — которые встретили и поняли
3. **Распечатайте** и повесьте рядом с компьютером
4. **Объясняйте термины внукам** — лучший способ запомнить!

**Помните:** Все эти "страшные" слова придумали обычные люди, чтобы называть простые вещи. Вы уже знаете большинство из них на практике!

---

Словарик будет пополняться по мере вашего продвижения. Сохраните его — он пригодится ещё не раз! 

## 2. Палитра цветов: Таблица с красивыми и понятными названиями цветов и их кодами.

Дорогой друг, цвета в CSS — это как краски в коробке художника! Но чтобы не путаться в тысячах оттенков, я собрал для вас самую полезную палитру — цвета с понятными названиями и тёплыми, "уютными" ассоциациями. Эти цвета идеально подходят для сайтов о хобби, семье, природе и всём, что дорого сердцу!

---



### Основная палитра — 24 самых полезных цвета

#### Тёплые, уютные цвета (для домашних, семейных сайтов)

Цвет (название)	Код (HEX)	RGB	Когда использовать	Пример настроения
Медовый	#FFD700	rgb(255, 215, 0)	Заголовки, акценты	Солнечный день, радость, тепло
Корица	#8B4513	rgb(139, 69, 19)	Текст, рамки, кнопки	Уют, домашний очаг, дерево
Ванильный крем	#FFFACD	rgb(255, 250, 205)	Фон страницы	Светло, мягко, нежно
Персиковый	#FFDAB9	rgb(255, 218, 185)	Фон блоков, карточек	Тёплый, дружелюбный, welcoming
Пыльная роза	#D8bfd8	rgb(216, 191, 216)	Акценты для женских тем	Нежность, романтика, весна
Терракотовый	#A0522D	rgb(160, 82, 45)	Кнопки, границы	Земля, природа, натуральность

## Природные, спокойные цвета (для сайтов о саде, хобби)

Цвет (название)	Код (HEX)	RGB	Когда использовать	Пример настроения
Лесная зелень	#228B22	rgb(34, 139, 34)	Заголовки, важные элементы	Сад, рост, природа, жизнь
Морская волна	#2E8B57	rgb(46, 139, 87)	Подзаголовки, ссылки	Свежесть, чистота, вода
Оливковый	#556B2F	rgb(85, 107, 47)	Фон для текста	Спокойствие, естественность
Небесная лазурь	#87CEEB	rgb(135, 206, 235)	Фон для блоков	Лёгкость, воздух, свобода
Песочный	#F4A460	rgb(244, 164, 96)	Фон страницы	Тепло, пляж, отдых
Изумрудный	#50C878	rgb(80, 200, 120)	Акценты, кнопки	Богатство, роскошь, драгоценности

## Нейтральные, элегантные цвета (для текста и фона)

Цвет (название)	Код (HEX)	RGB	Когда использовать	Пример настроения
Слоновая кость	#FFFFFF	rgb(255, 255, 240)	Основной фон	Чистота, элегантность, свет
Серебристый	#C0C0C0	rgb(192, 192, 192)	Границы, разделители	Современность, технологичность
Угольно-серый	#333333	rgb(51, 51, 51)	Основной текст	Читаемость, серьёзность
Бежевый	#F5F5DC	rgb(245, 245, 220)	Фон для карточек	Теплота, нейтральность
Мятный крем	#F5FFFA	rgb(245, 255, 250)	Фон для форм	Свежесть, чистота
Льняной	#FAF0E6	rgb(250, 240, 230)	Фон всего сайта	Ткань, рукоделие, натуральность

## Яркие, праздничные цвета (для акцентов, праздников)

Цвет (название)	Код (HEX)	RGB	Когда использовать	Пример настроения
Праздничный красный	#DC143C	rgb(220, 20, 60)	Важные кнопки, скидки	Энергия, страсть, срочность
Лавандовый	#E6E6FA	rgb(230, 230, 250)	Фон для цитат	Спокойствие, мечтательность
Абрикосовый	#FBCEB1	rgb(251, 206, 177)	Подсветка	Лето, фрукты, радость
Васильковый	#6495ED	rgb(100, 149, 237)	Ссылки, иконки	Доверие, надёжность
Вишнёвый	#DE3163	rgb(222, 49, 99)	Акценты в меню	Яркость, привлекательность
Шафрановый	#F4C430	rgb(244, 196, 48)	Предупреждения, выделения	Внимание, важность

## Тематические палитры — готовые наборы цветов для разных проектов

### Палитра 1: "Уютный домашний блог"

css

```
/* Тёплая, домашняя гамма */  
:root {  
    --фон: #FFFACD;      /* Ванильный крем */  
    --текст: #8B4513;    /* Корица */  
    --заголовки: #A0522D; /* Терракотовый */  
    --акценты: #FFD700;  /* Медовый */  
    --рамки: #D2B48C;    /* Загар */  
    --кнопки: #8B4513;   /* Корица */  
    --кнопки-текст: #FFFACD; /* Ванильный крем */  
}
```

**Идеально для:** Семейного блога, дневника, кулинарного сайта

## Палитра 2: "Сад моей мечты"

css

```
/* Природная, свежая гамма */  
:root {  
    --фон: #F5FFFA;      /* Мятный крем */  
    --текст: #2E8B57;    /* Морская волна */  
    --заголовки: #228B22; /* Лесная зелень */  
    --акценты: #50C878;  /* Изумрудный */  
    --рамки: #556B2F;    /* Оливковый */  
    --кнопки: #2E8B57;   /* Морская волна */  
    --кнопки-текст: #FFFFFF; /* Белый */  
}
```

**Идеально для:** Сайта о садоводстве, цветах, природе

## Палитра 3: "Вязание с любовью"

css

```
/* Мягкая, шерстяная гамма */  
:root {  
    --фон: #FAF0E6;      /* Льняной */  
    --текст: #556B2F;    /* Оливковый */  
    --заголовки: #8B4513; /* Корица */  
    --акценты: #D8BFD8;  /* Пыльная роза */  
    --рамки: #C0C0C0;    /* Серебристый */  
    --кнопки: #8B4513;   /* Корица */  
    --кнопки-текст: #FAF0E6; /* Льняной */  
}
```

**Идеально для:** Сайта о рукоделии, вязании, вышивке

## Палитра 4: "Семейный фотоархив"

css

```
/* Ностальгическая, винтажная гамма */  
:root {  
    --фон: #F5F5DC;      /* Бежевый */  
    --текст: #333333;    /* Угольно-серый */  
    --заголовки: #8B4513; /* Корица */  
    --акценты: #FFDAB9;  /* Персиковый */
```

```
--рамки: #8B4513; /* Корица */
--кнопки: #333333; /* Угольно-серый */
--кнопки-текст: #F5F5DC; /* Бежевый */
}
```

**Идеально для:** Семейного альбома, истории семьи, мемуаров

## Цвета по временам года (для сезонных сайтов)

### Весенняя палитра — пробуждение

Цвет	Код	Настроение	Использование
Цвет яблони	#FFB7C5	Нежность, начало	Фон, акценты
Свежая зелень	#90EE90	Рост, молодость	Кнопки, заголовки
Небесно-голубой	#87CEEB	Чистота, свежесть	Фон блоков
Фиалковый	#EE82EE	Творчество, вдохновение	Акценты
Лимонный	#FFF44F	Радость, энергия	Выделения

### Летняя палитра — яркость

Цвет	Код	Настроение	Использование
Лазурный	#007FFF	Море, свобода	Основной цвет
Песочный	#F4A460	Пляж, отдых	Фон
Арбузный	#FC6C85	Яркость, веселье	Акценты
Кукурузный	#FBEC5D	Солнце, тепло	Подсветка
Бирюзовый	#40E0D0	Свежесть, вода	Ссылки

### Осенняя палитра — уют

Цвет	Код	Настроение	Использование
Тыквенный	#FF7518	Тепло, урожай	Заголовки
Кленовый	#D2691E	Уют, домашний очаг	Основной цвет

Цвет	Код	Настроение	Использование
<b>Бордовый</b>	#800000	Глубина, зрелость	Акценты
<b>Золотой</b>	#FFD700	Богатство, изобилие	Выделения
<b>Шоколадный</b>	#7B3F00	Комфорт, надёжность	Текст

## Зимняя палитра — чистота

Цвет	Код	Настроение	Использование
<b>Снежно-белый</b>	#FFFafa	Чистота, ясность	Фон
<b>Ледяной синий</b>	#B0E0E6	Холод, свежесть	Акценты
<b>Серебристый</b>	#C0C0C0	Элегантность	Границы
<b>Еловый</b>	#2E8B57	Жизнь, надежда	Кнопки
<b>Сиреневый иней</b>	#E6E6FA	Нежность, мечтательность	Фон блоков



## Цвета для особых случаев

### День рождения, праздники

Событие	Основной цвет	Код	Дополнительные цвета
<b>День рождения</b>	Золотой	#FFD700	#DC143C (красный), #FFFFFF (белый)
<b>Новый год</b>	Красный	#DC143C	#228B22 (зелёный), #FFD700 (золотой)
<b>8 Марта</b>	Розовый	#FFC0CB	#90EE90 (зелёный), #FFFFFF (белый)
<b>Свадьба</b>	Кремовый	#FFFDD0	#D8BFD8 (сиреневый), #F5F5DC (бежевый)
<b>Юбилей</b>	Бордовый	#800000	#FFD700 (золотой), #FFFFFF (белый)

## Сезонные праздники

### Праздник Цветовая схема Коды

<b>Пасха</b>	Пастельная	#FFB7C5 (розовый), #90EE90 (зелёный), #87CEEB (голубой), #FFF44F (жёлтый)
<b>Масленица</b>	Яркая	#FFD700 (жёлтый), #DC143C (красный), #FF8C00 (оранжевый)
<b>Рождество</b>	Традиционная	#DC143C (красный), #228B22 (зелёный), #FFFFFF (белый), #FFD700 (золотой)
<b>Хэллоуин</b>	Контрастная	#FF7518 (оранжевый), #000000 (чёрный), #800080 (фиолетовый)

### Практическая таблица: "Какой цвет для чего"

#### Для текста (читаемость — главное!)

Тип текста	Рекомендуемый цвет	Код	Почему
<b>Основной текст</b>	Тёмно-серый	#333333	Не устают глаза, лучше читается
<b>Заголовки</b>	Тёмно-коричневый	#8B4513	Выделяется, но не кричит
<b>Ссылки</b>	Морская волна	#2E8B57	Видно, что это ссылка
<b>Ссылки при наведении</b>	Лесная зелень	#228B22	Понятно, что элемент активен
<b>Второстепенный текст</b>	Серый	#666666	Не отвлекает от главного

#### Для фона (создаём настроение)

Тип фона	Рекомендуемый цвет	Код	Эффект
<b>Фон всей страницы</b>	Ванильный крем	#FFFACD	Тёплый, уютный, не режет глаза
<b>Фон блока/карточки</b>	Бежевый	#F5F5DC	Выделяет блок, но не

Тип фона	Рекомендуемый цвет	Код	Эффект
	контрастно		
<b>Фон для важной информации</b>	Мятный крем	#F5FFFA	Привлекает внимание мягко
<b>Фон для меню</b>	Слоновая кость	#FFFFFF	Чисто, элегантно
<b>Фон для подвала</b>	Светло-серый	#F8F8F8	Отделяет от основного контента

## Для кнопок и интерактивных элементов

Тип кнопки	Цвет кнопки	Цвет текста	Код кнопки	Когда использовать
<b>Основная кнопка</b>	Корица	Ванильный крем	#8B4513	Главное действие (отправить, купить)
<b>Вторичная кнопка</b>	Морская волна	Белый	#2E8B57	Дополнительные действия
<b>Кнопка-призрак</b>	Прозрачная	Корица	transparent border: #8B4513	Не главные действия
<b>Опасная кнопка</b>	Мягкий красный	Белый	#DC143C	Удалить, отменить
<b>Выключенная кнопка</b>	Серебристый	Серый	#C0C0C0	Когда действие недоступно

## 🎯 Шпаргалка: Быстрый подбор цвета по настроению

Хочу создать чувство...

Чувство/ настроение	Основной цвет	Вспомогательные цвета
<b>Уюта и тепла</b>	Корица (#8B4513)	Ванильный крем (#FFFACD), Персиковый

Чувство/ настроение	Основной цвет	Вспомогательные цвета
		(#FFDAB9)
Свежести и природы	Лесная зелень (#228B22)	Мятный крем (#F5FFFA), Небесная лазурь (#87CEEB)
Элегантности	Угольно-серый (#333333)	Слоновая кость (#FFFFFF), Серебристый (#C0C0C0)
Радости и праздника	Медовый (#FFD700)	Праздничный красный (#DC143C), Лавандовый (#E6E6FA)
Спокойствия	Морская волна (#2E8B57)	Льняной (#FAF0E6), Бежевый (#F5F5DC)
Доверия	Васильковый (#6495ED)	Белый (#FFFFFF), Светло-серый (#F8F8F8)

## Практическое применение: Примеры кода с цветами

### Пример 1: Тёплый блог о семье

css

```
/* Используем "уютную" палитру */
body {
    background-color: #FFFACD; /* Ванильный крем - фон */
    color: #333333; /* Угольно-серый - основной текст */
    font-family: 'Georgia', serif;
}

h1, h2, h3 {
    color: #8B4513; /* Корица - заголовки */
    border-bottom: 2px solid #D2B48C; /* Загар - подчёркивание */
}

a {
    color: #2E8B57; /* Морская волна - ссылки */
    text-decoration: none;
```

```

}

a:hover {
    color: #228B22; /* Лесная зелень - при наведении */
    text-decoration: underline;
}

.button {
    background-color: #8B4513; /* Корица - кнопки */
    color: #FFFAF0; /* Ванильный крем - текст на кнопках */
    padding: 12px 24px;
    border-radius: 8px;
    border: none;
}

.card {
    background-color: #F5F5DC; /* Бежевый - фон карточек */
    border: 1px solid #D2B48C; /* Загар - рамка */
    border-radius: 10px;
    padding: 20px;
}

```

## Пример 2: Сайт о садоводстве

css

```

/* Используем "природную" палитру */
body {
    background-color: #F5FFFF; /* Мятный крем - фон */
    color: #2E8B57; /* Морская волна - текст */
    font-family: 'Arial', sans-serif;
}

h1 {
    color: #228B22; /* Лесная зелень - главный заголовок */
    text-shadow: 1px 1px 2px rgba(0,0,0,0.1);
}

.plant-card {
    background-color: #FFFFFF; /* Белый - фон карточки растения */
    border: 2px solid #90EE90; /* Светло-зелёный - рамка */
    border-radius: 12px;
    box-shadow: 0 4px 8px rgba(46, 139, 87, 0.1);
}

```

```
.important {  
    background-color: #FFFACD; /* Ванильный крем - выделение важного */  
    border-left: 5px solid #FFD700; /* Медовый - акцентная полоса */  
    padding: 15px;  
}
```

---

## Инструменты для работы с цветами

### Онлайн-генераторы (бесплатные):

1. [Colors.co](#) — нажимайте пробел, получайте красивые палитры
2. [ColorHunt.co](#) — готовые палитры от дизайнеров
3. [Adobe Color](#) — можно загрузить фото и получить цвета из него

### Как создать свою палитру:

1. **Выберите основной цвет** ( тот, что больше всего нравится)
2. **Добавьте 1-2 дополнительных** (похожих оттенка)
3. **Добавьте акцентный цвет** (контрастный, для кнопок и важного)
4. **Выберите нейтральные цвета** для текста и фона

### Правило 60-30-10:

- **60%** страницы — основной цвет (фон)
  - **30%** — вторичный цвет (блоки, карточки)
  - **10%** — акцентный цвет (кнопки, ссылки, важное)
- 



### Советы по выбору цветов для вашего сайта:

1. **Начинайте с любимого цвета** — то, что нравится вам, понравится и другим
2. **Смотрите на реальные объекты** — фото вашего сада, вязаных вещей, рецептов
3. **Используйте не более 4-5 цветов** на одной странице

- 4. Проверяйте контраст** — текст должен хорошо читаться на фоне
- 5. Учитывайте ассоциации** — зелёный = природа, синий = доверие, красный = энергия
- 6. Делайте перерывы** — если устали глаза, цвет слишком яркий
- 7. Спросите мнение** — покажите внукам, что они думают о цветах



## Шпаргалка для быстрого использования

### Скопируйте и используйте:

css

```
/* Тёплые цвета */  
.color-honey { color: #FFD700; } /* Медовый */  
.color-cinnamon { color: #8B4513; } /* Корица */  
.color-vanilla { color: #FFFACD; } /* Ванильный крем */  
.color-peach { color: #FFDAB9; } /* Персиковый */  
  
/* Природные цвета */  
.color-forest { color: #228B22; } /* Лесная зелень */  
.color-seagreen { color: #2E8B57; } /* Морская волна */  
.color-olive { color: #556B2F; } /* Оливковый */  
.color-skyblue { color: #87CEEB; } /* Небесная лазурь */  
  
/* Нейтральные */  
.color-ivory { color: #FFFFFF; } /* Слоновая кость */  
.color-charcoal { color: #333333; } /* Угольно-серый */  
.color-beige { color: #F5F5DC; } /* Бежевый */  
.color-linen { color: #FAF0E6; } /* Льняной */
```

### Для фонов:

css

```
.bg-honey { background-color: #FFD700; }  
.bg-cinnamon { background-color: #8B4513; }  
.bg-vanilla { background-color: #FFFACD; }  
.bg-forest { background-color: #228B22; }
```

---

**Дорогой друг, теперь у вас есть целая палитра красивых, проверенных цветов!** Не нужно гадать или подбирать наугад. Берите готовые сочетания, экспериментируйте, но главное — создавайте с удовольствием!

**Помните:** Самый лучший цвет — тот, который радует ваши глаза и подходит вашему проекту. Доверяйте своему вкусу!

Эта палитра будет вашим верным помощником во всех будущих проектах.

Сохраните её и обращайтесь к ней каждый раз, когда нужно выбрать цвета! 

### **3. Сборник готовых «рецептов»: Код для кнопки, красивой цитаты, рамки для фото – чтобы просто скопировать и использовать.**

Дорогой друг!

Теперь, когда вы уже многое знаете о CSS, настало время для самой приятной части — готовых решений. Представьте, что у вас на кухне есть набор проверенных рецептов: «Идеальные блины», «Быстрый суп», «Праздничный торт». Вы не придумываете их каждый раз с нуля — вы берёте готовый рецепт и, если нужно, немного адаптируете его под свой вкус.

Здесь собраны такие «рецепты» для веб-страниц. Вы можете просто скопировать код, вставить в свой файл style.css (или в тег <style>), и красивые элементы появятся на вашей странице как по волшебству! А если захочется что-то изменить — цвет, размер, отступы — вы теперь знаете, как это сделать.

---

#### **Рецепт 1: Универсальная дружелюбная кнопка**

Идеальна для ссылок «Подробнее», «Отправить», «Читать далее». Меняет цвет при наведении, как бы приглашая нажать.

#### **Как выглядит:**

(В браузере будет настоящая интерактивная кнопка)

#### **Код для CSS:**

```
css
/* Рецепт: Универсальная кнопка */
.моя-кнопка {
    display: inline-block;      /* Чтобы кнопке можно было задать отступы */
    padding: 14px 28px;        /* Внутренние отступы: сверху/снизу 14px, слева/справа 28px */
```

```

background-color: #4CAF50; /* Красивый зелёный цвет (можно изменить) */
color: white; /* Цвет текста на кнопке */
text-decoration: none; /* Убираем стандартное подчёркивание у ссылки */
font-size: 18px; /* Размер текста */
font-weight: bold; /* Текст жирный */
border-radius: 8px; /* Закругляем углы */
border: none; /* Убираем рамку (если это не ссылка, а <button>) */
cursor: pointer; /* Меняем курсор на "руку" при наведении */
text-align: center; /* Текст по центру */
transition: all 0.3s ease; /* Плавная анимация при изменении (цвета, тени) */
box-shadow: 0 4px 8px rgba(0, 0, 0, 0.1); /* Лёгкая тень снизу */

}

/* Эффект при наведении курсора */
.moya-knopka:hover {
    background-color: #45a049; /* Чуть темнее зелёный */
    box-shadow: 0 6px 12px rgba(0, 0, 0, 0.15); /* Тень становится больше */
    transform: translateY(-2px); /* Кнопка слегка "подпрыгивает" вверх */
}

/* Эффект при нажатии */
.moya-knopka:active {
    transform: translateY(0); /* Возвращаем на место */
    box-shadow: 0 2px 4px rgba(0, 0, 0, 0.1); /* Тень уменьшается */
}

```

## Как использовать в HTML:

html

```

<!-- Если это ссылка (куда-то ведёт) -->
<a href="#" class="moya-knopka">Подробнее</a>

<!-- Если это кнопка (для действия на странице) -->
<button class="moya-knopka">Отправить сообщение</button>

```

## Как настроить под себя:

- Цвет кнопки:** Поменяйте #4CAF50 на любой другой HEX-код (например, #3498db для синего).
  - Размер:** Измените цифры в padding (больше — кнопка крупнее) и font-size.
  - Скругление:** Увеличьте border-radius (например, до 20px) для более округлой кнопки.
-

## Рецепт 2: Элегантная рамка для фотографии

Превратит обычное изображение в акцентный элемент со скруглёнными углами и лёгкой тенью, как в современном фотоальбоме.

### Как выглядит:

### Код для CSS:

css

```
/* Рецепт: Рамка для фотографии */
.фото-в-рамке {
    display: block;          /* Чтобы картинка не вела себя как текст */
    max-width: 100%;         /* Картина никогда не вылезет за пределы блока */
    height: auto;            /* Высота подстраивается автоматически */
    padding: 12px;           /* Внутренний отступ — будет фон вокруг фото */
    background-color: white; /* Цвет "паспарту" внутри рамки */
    border: 8px solid #f1f1f1; /* Широкая светлая рамка */
    border-radius: 12px;      /* Скругляем внешние углы рамки */
    box-shadow: 0 10px 25px rgba(0, 0, 0, 0.1); /* Мягкая объёмная тень */
    margin: 20px auto;        /* Отступы сверху/снизу и выравнивание по центру */
    transition: transform 0.3s ease; /* Плавное увеличение при наведении */
}

/* Эффект при наведении: фото слегка увеличивается */
.фото-в-рамке:hover {
    transform: scale(1.02); /* Увеличиваем на 2% */
}
```

### Как использовать в HTML:

html

```

```

### Как настроить под себя:

- Цвет рамки:** Поменяйте `#f1f1f1` на другой (например, `#e6d5c3` для бежевого).
- Толщина рамки:** Измените `8px` на `4px` для тонкой или `15px` для массивной.

3. **Скругление:** `border-radius: 50%` сделает фото круглым!
  4. **Размер:** Задайте фиксированную ширину, например: `width: 300px;`
- 

## Рецепт 3: Красивая цитата (выделенный блок)

Для важных мыслей, мудрых изречений или просто чтобы красиво оформить чьи-то слова.

### Как выглядит:

«Садоводство — это искусство, где холст — земля, а краски — живые цветы. Каждый новый бутон — это маленькое чудо, напоминающее нам, что красота требует времени, терпения и любви.»

### Код для CSS:

css

```
/* Рецепт: Блок для красивой цитаты */  
.моя-цитата {  
    font-family: 'Georgia', serif; /* Классический шрифт с засечками */  
    font-size: 20px;             /* Чуть крупнее основного текста */  
    line-height: 1.6;            /* Удобное расстояние между строками */  
    color: #555;                /* Мягкий тёмно-серый, не чёрный */  
    background-color: #f9f9f9;   /* Светлый фон */  
    padding: 30px 40px;          /* Просторные внутренние отступы */  
    margin: 30px 0;              /* Отступы сверху и снизу от других элементов */  
    border-left: 6px solid #8b4513; /* Толстая коричневая полоса слева */  
    border-radius: 0 8px 8px 0;   /* Скругляем только правые углы */  
    box-shadow: 3px 3px 10px rgba(0, 0, 0, 0.05); /* Едва заметная тень */  
    font-style: italic;          /* Текст курсивом */  
    position: relative;         /* Нужно для иконки кавычек (опционально) */  
}  
  
/* Необязательно: Добавим декоративные кавычки */  
.моя-цитата::before {  
    content: "";               /* Открывающая кавычка */  
    font-size: 80px;            /* Очень большая */  
    color: rgba(139, 69, 19, 0.1); /* Полупрозрачный коричневый */
```

```
position: absolute;  
top: 10px;  
left: 20px;  
font-family: 'Times New Roman', serif;  
}
```

## Как использовать в HTML:

html

```
<blockquote class="моя-цитата">
```

Садоводство — это искусство, где холст — земля, а краски — живые цветы. Каждый новый бутон — это маленькое чудо, напоминающее нам, что красота требует времени, терпения и любви.

```
</blockquote>
```

## Как настроить под себя:

- Цвет акцента:** Поменяйте #8b4513 (коричневый) на #2e8b57 (зелёный) или #3498db (синий).
  - Фон:** Сделайте фон белым (#fff) или очень светлым пастельным (#ffffaf).
  - Шрифт:** Подключите красивый шрифт с Google Fonts (например, 'Playfair Display') и укажите его в font-family.
- 

## Рецепт 4: Простая горизонтальная навигация (меню)

Создаёт аккуратную панель ссылок вверху страницы — основу для навигации по вашему сайту.

## Как выглядит:

Главная | Обо мне | Моё хобби | Фотогалерея | Контакты

## Код для CSS:

css

```
/* Рецепт: Простое горизонтальное меню */
```

```

.моё-меню {
    background-color: #333;      /* Тёмный фон меню */
    padding: 0;                  /* Убираем стандартные отступы */
    margin: 0;                   /* Убираем внешние отступы */
    list-style-type: none;       /* Убираем маркеры (точки) у списка */
    display: flex;              /* Волшебное свойство: выстраиваем пункты в ряд */
    justify-content: center;    /* Выравниваем пункты по центру */
    border-radius: 8px;          /* Слегка скругляем углы всего меню */
    flex-wrap: wrap;            /* Если не помещается — переносит на новую строку */
}

/* Стиль для каждого пункта меню */
.моё-меню li {
    margin: 0;                  /* Убираем отступы между пунктами */
}

/* Стиль для ссылок внутри меню */
.моё-меню a {
    display: block;             /* Ссылка занимает всю площадь пункта */
    color: white;               /* Белый текст */
    text-decoration: none;       /* Без подчёркивания */
    padding: 18px 25px;          /* Просторные кликабельные области */
    font-size: 18px;
    font-weight: 500;             /* Полужирный */
    transition: background-color 0.3s ease; /* Плавная смена фона */
}

/* Эффект при наведении на ссылку */
.моё-меню a:hover {
    background-color: #4CAF50;   /* Зелёный фон при наведении */
    border-radius: 4px;           /* Слегка скругляем углы у активного элемента */
}

```

## Как использовать в HTML:

```

html
<ul class="моё-меню">
    <li><a href="index.html">Главная</a></li>
    <li><a href="about.html">Обо мне</a></li>
    <li><a href="hobby.html">Моё хобби</a></li>
    <li><a href="gallery.html">Фотогалерея</a></li>
    <li><a href="contact.html">Контакты</a></li>
</ul>

```

## Как настроить под себя:

1. **Цвета:** Поменяйте #333 (фон меню) и #4CAF50 (цвет при наведении).
  2. **Выравнивание:** Замените justify-content: center; на flex-start (к левому краю) или flex-end (к правому краю).
  3. **Вертикальное меню:** Замените display: flex; на display: block; в классе .моё-меню и уберите justify-content.
- 

## Важная подсказка по использованию «рецептов»

1. **Копируйте код** в ваш файл style.css.
2. **В HTML-файле** добавьте элементам нужный класс, как показано в примерах.
3. **Не бойтесь экспериментировать!** Меняйте цвета, отступы, размеры. Вы ничего не сломаете, а только научитесь лучше чувствовать CSS.
4. **Сохраняйте свои удачные варианты** в отдельном файле (например, мои-рецепты.css). Это будет ваша личная библиотека дизайна!

**Запомните:** Эти рецепты — не догма, а отправная точка. Самые красивые решения рождаются, когда вы адаптируете готовое под свой уникальный вкус и задачу. Удачи в творчестве!

---

Следующий раздел: «4. Список добрых и бесплатных ресурсов» — где брать шрифты, иконки, картинки для ваших проектов.

---

## **4. Список добрых и бесплатных ресурсов: Где брать шрифты, иконки, картинки.**

Дорогой друг!

Теперь, когда вы умеете создавать красоту с помощью CSS, настало время наполнить ваши страницы качественным «содержимым»: красивыми шрифтами, понятными иконками и яркими фотографиями. Хорошая новость: в интернете есть огромное количество **бесплатных, легальных и очень качественных** ресурсов, созданных талантливыми людьми для всех.

Здесь — ваш личный «адресный столик». Эти сайты — как проверенные магазины, где всё можно брать смело, не нарушая законов и не беспокоясь о качестве.

---



### **Правило безопасности и уважения**

Прежде чем идти «за покупками», запомните одно золотое правило:

**| Всегда проверяйте условия использования (License)!**

Обычно нужна **бесплатная лицензия для некоммерческого использования (Free for personal use)**. Это значит, что вы можете использовать ресурс для своего сайта-хобби, семейного альбома или поздравительной открытки, но не для продажи товаров или рекламы бизнеса. Часто авторы просто просят указать их имя где-то на странице.

#### **Где искать эту информацию?**

Обычно на сайте есть кнопка или ссылка: License, Лицензия, Terms of Use, Условия использования. Часто нужный нам тип лицензии так и называется: **CC0 (Public Domain), Creative Commons, MIT License, SIL Open Font License**. Если видите эти слова — можно смело использовать.

---

A  
B  
C  
D

## 1. Где брать красивые шрифты

Шрифт — это «одежда» вашего текста. Вот лучшие бесплатные гардеробные:

### A. Google Fonts — самый простой и надёжный друг

- Ссылка: [fonts.google.com](https://fonts.google.com)
- Почему он первый?
  - Очень простой интерфейс с поддержкой русского языка.
  - Все шрифты абсолютно бесплатны для любого использования (даже коммерческого).
  - Автоматическая техническая настройка. Сайт сам даст вам готовый код для вставки в HTML и CSS. Вы просто выбираете шрифт, нажимаете «Select this style», копируете код из появившейся панели справа — и всё работает!
  - Есть фильтр «Cyrillic». Обязательно включите его, чтобы видеть только шрифты, поддерживающие русские буквы (кириллицу).
- Что искать: Для заголовков — Playfair Display, Lora. Для основного текста — Roboto, Open Sans. Для элегантности — Cormorant Garamond.

### B. Font Squirrel — коллекция качественных бесплатных шрифтов

- Ссылка: [www.fontsquirrel.com](http://www.fontsquirrel.com)
- Чем хорош? Здесь тщательно отбирают действительно качественные и профессиональные шрифты. Все они бесплатны для коммерческого использования. Отличное место, чтобы найти что-то уникальное.
- Важно: Также используйте фильтр «Cyrillic / Russian».

#### Памятка по выбору шрифта:

1. Читаемость — прежде всего. Для длинных текстов выбирайте простые, неброские шрифты без засечек (Sans-serif) или с аккуратными засечками (Serif).
2. Не больше 2-3 шрифтов на страницу. Один — для заголовков, второй — для основного текста, третий (если нужен) — для особых акцентов.

3. **Проверьте на кириллице!** Напишите в окне предпросмотра русскую фразу: «Съешь же ещё этих мягких французских булок, да выпей чаю». Увидите все буквы.

---

## 2. Где брать иконки и небольшие картинки (иллюстрации)

Иконки — это маленькие символы, которые делают интерфейс понятным: домик для «Главной», конверт для «Контактов», стрелочки, галочки.

### A. FontAwesome — самая популярная библиотека иконок

- **Ссылка:** [fontawesome.com](http://fontawesome.com)
- **Почему он любим всеми?** Это не просто картинки, а **шрифт**. Вы подключаете его к странице, а потом вставляете иконки как буквы специальным кодом. Это очень удобно: иконки можно красить в любой цвет CSS-ом, менять размер, добавлять тень.
- **Как начать:** Зайдите в раздел «Start for Free». Вам предложат бесплатный набор из тысяч иконок. Скопируйте ссылку для подключения (как с Google Fonts) и используйте иконки по имени.

### B. Feather Icons — простые и изящные outline-иконки

- **Ссылка:** [feathericons.com](http://feathericons.com)
- **Особенность:** Минималистичные, контурные, очень современные иконки. Все в одном стиле. Можно скачать по одной или весь набор. Лицензия: MIT (очень свободная).

### C. Flaticon — огромная коллекция иконок в разных стилях

- **Ссылка:** [www.flaticon.com](http://www.flaticon.com)
- **Почему сюда?** Миллионы иконок на любую тему! Есть бесплатный лимит скачиваний в день. Обязательно смотрите, нужно ли указывать автора (требование лицензии будет написано рядом с кнопкой скачивания).

## D. Unsplash — фотографии, которые выглядят как иконки (и не только)

- **Ссылка:** [unsplash.com](https://unsplash.com)
  - **Что это?** В основном, это сайт с фото. Но здесь можно найти прекрасные текстуры, абстрактные фоны и простые предметные фото на белом фоне, которые отлично работают как большие, атмосферные «иконки».
- 

## 3. Где брать красивые бесплатные фотографии

Ваши личные фотографии — это самое ценное. Но для фона, иллюстраций или если нужен конкретный пейзаж, пригодятся эти чудесные хранилища.

**Важно!** На этих сайтах фотографии с лицензией **CC0** — это значит, автор передал их в общественное достояние. Вы можете использовать их где угодно, даже не указывая автора (хотя это всегда хороший тон — поблагодарить).

### Топ-3 сайта, где всё бесплатно и без заморочек:

1. **Unsplash** ([unsplash.com](https://unsplash.com)) — король бесплатных фото. Невероятно красивые, художественные снимки от фотографов со всего мира. Качество — как в глянцевых журналах. Просто найдите нужное слово (например, «garden», «knitting», «family») на английском и скачивайте.
2. **Pexels** ([pexels.com](https://pexels.com)) — ещё одна гигантская, очень дружелюбная коллекция. Есть не только фото, но и бесплатные видеозаставки. Удобный поиск.
3. **Pixabay** ([pixabay.com](https://pixabay.com)) — здесь есть всё: фото, иллюстрации, векторная графика, иконки, видео. Огромный выбор. Интерфейс переведён на русский язык.

### Как искать фотографии эффективно?

- **Ищите на английском.** Базы международные, поэтому ключевые слова лучше вводить по-английски: grandmother, garden, tea, book, nature, family, craft, flower.

- **Используйте цвета.** На многих сайтах есть фильтр по цвету. Хотите сделать сайт в голубых тонах? Найдите голубые фото для фона.
  - **Скачивайте самый большой размер.** Так вы сможете использовать фото и как фон, и как небольшую иллюстрацию без потери качества.
- 

## ✨ 4. Дополнительные сокровищницы

### Для фонов и текстур:

- **Subtle Patterns** ([www.toptal.com/designers/subtlepatterns](http://www.toptal.com/designers/subtlepatterns)) — коллекция изящных, ненавязчивых текстур в виде маленьких повторяющихся картинок (tile). Идеальны для фона сайта вместо однотонного цвета.

### Для цветовых палитр:

- **Colors** ([colors.co](http://colors.co)) — генератор гармоничных цветовых схем. Нажимайте пробел — получаете новую палитру. Копируйте HEX-коды и используйте в своём CSS.
- **Color Hunt** ([colorhunt.co](http://colorhunt.co)) — готовые, подобранные дизайнерами палитры из 4 цветов. Бесконечный источник вдохновения.

### Для вдохновения (посмотреть, как сделано у других):

- **CodePen** ([codepen.io](http://codepen.io)) — это социальная сеть для веб-разработчиков. Здесь люди выкладывают свои эксперименты с HTML, CSS и JavaScript. Можно найти потрясающие идеи, а главное — **посмотреть и скопировать код**, чтобы разобраться, как это сделано. Ищите по тегам: button, card, menu.
- 



### Ваш порядок действий, когда нужно что-то найти:

1. **Определите, что вам нужно:** Шрифт для заголовка? Иконка «сердечко»? Фотография осеннего леса для фона?
2. **Идите в нужный раздел этого списка** и откройте рекомендованный сайт.
3. **Воспользуйтесь поиском** (помня про английские слова).

- 4. Проверьте лицензию** (обычно всё на рекомендованных сайтах уже безопасно).
- 5. Скачайте файл** или скопируйте код для подключения.
- 6. Добавьте ресурс в свой проект**, как мы учились в предыдущих главах.

**Помните:** Интернет-сообщество дизайнеров и фотографов щедро делится своими работами. Используя эти ресурсы, вы становитесь частью этого творческого круга. А когда-нибудь, возможно, и вы сможете поделиться своими наработками с другими!

Удачи в наполнении вашего цифрового дома красотой! Ваш сайт станет по-настоящему уникальным и душевным.

---

Этот раздел завершает наши «Полезные шпаргалки». Вы теперь вооружены не только знаниями, но и лучшими инструментами. Творите с удовольствием!