

"Создаём свой первый веб-сайт"

Оглавление:

Предисловие: Для кого эта книга и как её читать

Не бойтесь компьютера: это всего лишь инструмент.
Наш главный принцип: "От простого — к сложному".
Что Вам понадобится: только компьютер и обычное приложение
"Блокнот".
Как построены уроки: теория → пример → практическое задание.

Часть 1: Первые шаги. Знакомство с миром веб-страниц

Глава 1: Введение в мир Интернета и веб-страниц

- 1.1. Что такое Интернет и Всемирная паутина (простыми словами).
- 1.2. Как браузер (Chrome, Edge) показывает нам страницы?
- 1.3. Что такое HTML? Не язык программиста, а "язык разметки" — как инструкция для браузера.
- 1.4. Практика: открываем браузер и смотрим на любую страницу "глазами компьютера" (используем инструмент "Просмотр кода").

Глава 2: Подготовка рабочего места

- 2.1. Создаём свою рабочую папку на компьютере "Мой первый сайт".
- 2.2. Наш главный инструмент — Блокнот (или его улучшенная версия Notepad++). Почему не Word?
- 2.3. Создаём первый файл с правильным именем: index.html¹
- 2.4. Практика: создаём папку и файл, открываем его в Блокноте и в браузере.

¹ Все названия веб-страниц пишутся только английскими буквами и словами!!!

Часть 2:

Основы HTML. Пишем первую страницу

Глава 3:

Структура каждой HTML-страницы

- 3.1. Обязательные теги: `<html>`, `<head>`, `<body>`. Кто за что отвечает?
- 3.2. Заголовок страницы `<title>`: что мы видим во вкладке браузера.
- 3.3. Простая аналогия: страница — это человек (`<html>`). Голова (`<head>`) — для мыслей (название), тело (`<body>`) — для содержимого.
- 3.4. Практика: пишем "скелет" нашей первой страницы.

Глава 4:

Добавляем текст на страницу

- 4.1. Теги для заголовков: `<h1>`, `<h2>`, `<h3>` (как заголовки в газете).
- 4.2. Абзац текста: тег `<p>`.
- 4.3. Перенос строки `
`: когда нужно просто перейти на новую строку.
- 4.4. Горизонтальная линия `<hr>` для разделения содержимого.
- 4.5. Практика: создаём страницу-визитку о себе с заголовком и несколькими абзацами.

Глава 5:

Простое оформление текста

- 5.1. Жирный шрифт `` или ``.
- 5.2. Курсив `<i>` или ``.
- 5.3. Почему есть два варианта? Разница между внешним видом и важностью.
- 5.4. Практика: выделяем важные мысли на своей визитке.

Часть 3:

Создаём современную и удобную страницу

Глава 6:

Списки — для порядка и структуры

- 6.1. Маркированный список `` и `` (для перечня чего угодно).
- 6.2. Нумерованный список `` и `` (для инструкций или рецептов).
- 6.3. Практика: добавляем на страницу список своих увлечений или список дел на день.

Глава 7: **Ссылки — "двери" в другие страницы**

- 7.1. Волшебный тег `<a>` и его атрибут `href`.
- 7.2. Как сделать ссылку на другой сайт (например, на любимый новостной портал).
- 7.3. Как сделать ссылку на другую свою страницу.
- 7.4. Практика: создаём вторую страницу "Мои любимые сайты" и связываем её с главной ссылками.

Глава 8: **Картинки — чтобы страница "заиграла"**

- 8.1. Тег ``: атрибуты `src` (адрес картинки) и `alt` (описание для тех, кто не видит).
- 8.2. Где найти простые и бесплатные картинки для тренировки.
- 8.3. Как правильно сохранить картинку в свою рабочую папку.
- 8.4. Практика: добавляем свою фотографию или красивую картинку на главную страницу.

Глава 9: **Простые таблицы (для наглядности)**

- 9.1. Создаём таблицу: теги `<table>`, `<tr>` (строка), `<td>` (ячейка).
- 9.2. Заголовки столбцов в таблице: тег `<th>`.
- 9.3. Простая рамка для таблицы.
- 9.4. Практика: создаём таблицу "Расписание моих занятий" или "Расходы за неделю".

Часть 4: **Важные мелочи и публикация сайта**

Глава 10: **Комментарии в коде — "записки для себя"**

- 10.1. Что такое комментарий `<!-- -->` и зачем он нужен.
- 10.2. Как комментарии помогают не запутаться в своём коде.

10.3. Практика: добавляем комментарии в свой код, чтобы объяснить, что делает каждый раздел.

Глава 11: Формы — для общения с посетителями

- 11.1. Простая форма <form>.
- 11.2. Поле для ввода текста <input type="text">.
- 11.3. Кнопка "Отправить" <input type="submit">.
- 11.4. Большое поле для текста <textarea> (для отзывов или писем).
- 11.5. Практика: создаём на странице "Гостевая книга" форму, где посетители могут оставить Вам сообщение.

Глава 12: Проверяем свою работу и исправляем ошибки

- 12.1. Что такое "валидатор" и зачем он нужен? (Простая аналогия с проверкой орфографии).
- 12.2. Простой онлайн-инструмент для проверки HTML-кода.
- 12.3. Как читать сообщения об ошибках и не пугаться их.
- 12.4. Практика: проверяем свою страницу и исправляем найденные ошибки.

Глава 13: Как показать свой сайт миру?

- 13.1. Что такое хостинг и домен (простыми словами).
- 13.2. Обзор бесплатных хостингов (например, GitHub Pages — объяснение максимально просто).
- 13.3. Простая инструкция: как загрузить файлы своего сайта в Интернет.
- 13.4. Ваш сайт в сети! Дайте ссылку детям и друзьям.

Приложения

Приложение А: Шпаргалка по основным тегам HTML.

Удобная таблица для быстрого поиска.

Приложение Б: Словарик терминов.

Объяснение всех непонятных слов (браузер, тег, атрибут, хостинг и т.д.).

Приложение В: Что учить дальше?

Краткий обзор CSS (для красивого оформления) и где найти такие же простые уроки по нему.

Приложение Г: Ответы к практическим заданиям.

Примеры готовых кодов для самопроверки.

**Предисловие:
Для кого эта книга и как её читать**

**Не бойтесь компьютера:
это всего лишь инструмент**

1. Введение: Откуда берётся страх?

"Я его случайно сломаю", "Нажму не на ту кнопку, и всё исчезнет", "Это слишком сложно для моего возраста" — эти мысли знакомы многим, кто только начинает осваивать компьютер. Давайте сразу договорим: этот страх абсолютно нормален и естественен. Он возникает из-за незнакомой среды, как если бы Вы впервые сели за руль автомобиля или взяли в руки новый кухонный гаджет.

Но есть и хорошая новость: **с компьютером работать гораздо безопаснее, чем с тем же автомобилем**. Самое страшное, что может случиться — Вам придётся перезагрузить программу или, в крайнем случае, сам компьютер. Ваш компьютер не взорвётся, не сломается окончательно от одного неверного клика и не причинит Вам физического вреда.

2. Проведём аналогии: Компьютер — это Ваш старый знакомый

Давайте посмотрим на компьютер не как на загадочный ящик, а как на обычный бытовой инструмент, которых в Вашей жизни было множество.

Компьютер — это как швейная машинка.

Вы же не боитесь швейной машинки? Вы знаете, что есть нитка, иголка, есть педаль, которая регулирует скорость. Вы учились заправлять нитку, подбирать ткани. С компьютером — то же самое! Есть программа (как функция машинки — прямая строчка, зигзаг), есть кнопки "Сохранить", "Напечатать" (как педаль и колесо намотки). Вы будете изучать его функции по одной, шаг за шагом.

Компьютер — это как кухонный комбайн.

У него есть разные насадки (ножи, тёрки) для разных задач. Вы не пытаетесь использовать все насадки сразу, чтобы приготовить суп. Вы берёте одну — нож, чтобы нарезать овощи. В компьютере программы — это и есть эти "насадки". Браузер — для интернета, Блокнот — для

написания текста, медиаплеер — для музыки. Вы используете только то, что нужно в данный момент.

Создание сайта на HTML — это как вязание по схеме.

Вы помните, как вяжете по схеме, где есть условные обозначения: лицевая петля, изнаночная, накид? Теги HTML — это те же условные обозначения! `<p>` — это "начать новый абзац", `` — это "сделать следующие буквы жирными". Вы не придумываете эти правила, Вы просто следуете им, чтобы получить готовое "изделие" — веб-страницу.

3. Ваша безопасность: Правила, которые защитят вас

Чтобы окончательно развеять страхи, запомните три "железных" правила:

Правило 1: Сохранение — Ваша лучшая страховка.

Когда Вы пишете текст или, как в нашем случае, HTML-код, **помимо нажимайте Ctrl+S (это комбинация клавиш для сохранения)**. Это как делать пометки в блокноте карандашом, который не стирается. Даже если что-то пойдёт не так, у Вас останется сохранённая копия Вашей работы. Мы научимся это делать в одной из первых глав.

Правило 2: Вы ничего не сломаете.

Работая с HTML-файлами в Блокноте, Вы имеете дело с **копиями** системных файлов компьютера. Вы не можете удалить или повредить Windows, просто написав неправильный тег в своём файле `about_me.html`. Это всё равно что испортить пирог на кухне — это не сломает саму духовку.

Правило 3: Всё можно исправить.

Сделали ошибку в коде? Страница отображается криво? **Ничего страшного!** Просто вернитесь в Блокнот, найдите эту ошибку (мы научимся это делать) и исправьте её. Это как переписать слово в письме. Компьютер не станет Вас ругать, он просто терпеливо будет ждать, когда Вы всё поправите.

4. Цель — не стать программистом, а получить новый инструмент

Вам не нужно запоминать тысячи команд. Наша цель — научиться пользоваться одним конкретным инструментом — HTML — для решения одной конкретной и интересной задачи: создания своей собственной веб-страницы.

Это увлекательное занятие, которое:

Тренирует память и внимание (как разгадывание кроссвордов).

Даёт творческий результат — Вы своими руками создаёте что-то о себе, своих увлечениях, воспоминаниях.

Помогает лучше понимать тот мир, в котором живут Ваши дети и внуки.

Заключение:

Отнеситесь к компьютеру как к умному, но бездушному помощнику. Он не умеет злиться, обижаться или смеяться над ошибками. Он всего лишь выполняет Ваши команды. Ваша задача — научиться давать ему эти команды на понятном ему языке. И этот язык — HTML — мы с Вами начнём изучать прямо сейчас.

Так что глубоко вдохните, улыбнитесь и помните: Вы управляете инструментом, а не он Вами.

Наш главный принцип: "От простого — к сложному"

1. Почему этот принцип так важен именно для нас?

Когда мы осваиваем что-то совершенно новое, особенно в зрелом возрасте, главная опасность — это перегрузить себя информацией и потерять уверенность. Наш мозг, как и мышцы, нуждается в постепенной тренировке. Мы не начинаем занятия физкультурой с марафонского забега, а сначала делаем легкую разминку.

Этот принцип — наш надежный проводник в мире HTML. Он гарантирует, что на каждом этапе Вы будете:

Понимать, что Вы делаете и зачем.

Видеть немедленный результат своих действий.

Чувствовать уверенность в своих силах перед переходом на следующий уровень.

2. Как это будет работать на практике? Конкретные примеры из нашего учебника

Давайте проследим, как этот принцип будет реализован в нашем обучении:

Пример 1: Изучение тегов

Шаг 1 (Простое): Мы начнем с тега `<p>`. Вы напишете один абзац текста, сохраните файл и откроете его в браузере. Результат: Вы видите свой текст на экране. Успех!

Шаг 2 (Немного сложнее): Мы добавим заголовок `<h1>` перед абзацем. Вы увидите, как текст становится крупнее и жирнее. Вы понимаете связь: тег → изменение внешнего вида.

Шаг 3 (Еще сложнее): Мы соединим несколько тегов: создадим страницу с заголовком `<h1>`, двумя разными подзаголовками `<h2>` и несколькими абзацами `<p>`. Вы уже не просто используете теги, а структурируете документ.

Пример 2: Работа со ссылками и изображениями

Шаг 1 (Простое): Мы научимся делать ссылку на другой сайт, например, на сайт любимой газеты. Вы просто копируете адрес и вставляете его в тег <a>.

Шаг 2 (Сложнее): Потом мы создадим второй HTML-файл в нашей папке (например, hobby.html) и сделаем ссылку с главной страницы на эту новую. Здесь Вы уже понимаете связь между файлами.

Шаг 3 (Еще сложнее): Позже мы добавим изображение на страницу hobby.html и, может быть, даже сделаем это изображение ссылкой. Вы комбинируете изученные элементы в новую конструкцию.

Пример 3: Процесс создания всего сайта

Урок 1: Создадим один-единственный файл index.html с заголовком и абзацем.

Через несколько уроков: У нас будет 3-4 связанных страницы (о себе, мои увлечения, контакты) с текстом, картинками и списками.

К концу учебника: Мы добавим на страницу "Контакты" простую форму для обратной связи.

3. Что это значит для вас, как для ученика?

Никакого информационного "шторма". Мы будем вводить не более 1-2 новых понятий за один урок. Вы не столкнетесь с десятком незнакомых тегов на первом же занятии.

Прочное усвоение. Прежде чем двигаться дальше, мы будем закреплять материал на практических заданиях. Каждый новый "кирпичик" знаний будет ложиться на прочный "фундамент" из уже усвоенного.

Мотивация и уверенность. В конце каждого занятия у Вас на руках будет готовый, работающий результат. Вы не просто "прошли тему", а создали что-то конкретное. Это чувство достижения — лучший двигатель прогресса.

Право на ошибку. Этот принцип напоминает нам, что не нужно пытаться обнять необъятное. Если Вы не поняли что-то сложное, значит, нужно вернуться на шаг назад и повторить простое. Это нормально и правильно.

4. Чего мы НЕ будем делать:

Мы не будем с первого урока говорить о сложных современных стандартах HTML5.

Мы не будем углубляться в теорию и историю развития интернета.

Мы не будем изучать несколько способов сделать одно и то же. Мы выберем один, самый простой и понятный, и будем использовать его.

Заключение:

Принцип "*От простого — к сложному*" — это наше обещание Вам, что путь обучения будет комфортным, предсказуемым и приносящим радость. Мы будем подниматься по лестнице знаний медленно, прочно ставя ногу на каждую ступеньку. Не нужно перепрыгивать через три ступеньки сразу — мы дойдем до вершины уверенным шагом, и вид оттуда будет прекрасным.

Что Вам понадобится: только компьютер и обычный Блокнот

1. Введение: Минимализм — наше преимущество

Одна из главных причин, по которой люди откладывают обучение чему-то новому в цифровой сфере — это кажущаяся сложность подготовки. Кажется, что нужно устанавливать десятки программ, разбираться в их настройках, искать ошибки и т.д. Мы ломаем этот стереотип. Для создания Вашего первого (и даже не первого!) сайта Вам не нужны дорогие и сложные программы. Всё необходимое уже есть в Вашем компьютере по умолчанию. Это демократично и снимает психологический барьер.

2. Детальный разбор необходимого

А) Компьютер

Любая модель и мощность: Подойдет абсолютно любой компьютер — стационарный РС или ноутбук, неважно какой марки (HP, Lenovo, Acer, Asus и т.д.).

Любая операционная система:

Если у Вас Windows (7, 8, 10, 11) — идеально, мы будем ориентироваться на нее.

Если у Вас Mac — тоже прекрасно. Вместо Блокнота там будет программа "*TextEdit*", а принципы те же.

Требования: Вашему компьютеру не нужно быть мощным. HTML-файлы — это простые текстовые файлы, они весят очень мало и не требуют для обработки почти никаких ресурсов. Компьютеру 10-летней давности с этим справится так же легко, как и новому.

Б) Браузер

Что это? Это программа, через которую Вы выходите в интернет (например, Google Chrome, Яндекс.Браузер, Microsoft Edge, Mozilla Firefox).

Зачем он нужен? Браузер — это наш "зрительный зал". В нем мы будем смотреть, что получилось из нашего HTML-кода. Мы пишем "пьесу" в Блокноте, а "ставят её на сцене" в браузере.

Какой выбрать? Подойдет любой. Рекомендую иметь под рукой два (например, Chrome и Edge), чтобы проверять, что Ваша страница везде выглядит одинаково.

В) Текстовый редактор (главный инструмент)

Обычный Блокнот (Notepad)

Где найти? Он есть в любой версии Windows. Находится через "Пуск" -> "Все программы" -> "Стандартные" -> "Блокнот".

Почему он?

Простота: В нем нет лишних кнопок, меню и функций, которые могут нас запутать. Он делает только одно — редактирует текст.

Наглядность: Мы пишем чистый код, без подсказок и автоматических дополнений. Это поможет нам лучше понять и запомнить каждую "букву" языка HTML.

Надежность: Он никогда не "зависнет" и не "сломается" от простого текста.

Улучшенная альтернатива: Notepad++

Что это? Бесплатная программа, которую можно легко скачать из интернета. Это такой же текстовый редактор, но "с подсветкой синтаксиса".

В чем преимущество? Он автоматически подсвечивает теги HTML разными цветами. Например, теги будут синими, текст — черным, а атрибуты — красными. Это очень помогает глазам находить ошибки.

Обязательно ли его ставить? НЕТ! Это просто приятное удобство. Мы начнем с обычного Блокнота, чтобы прочувствовать основы. Позже, если захотите, Вы легко установите Notepad++².

3. Чего нам НЕ понадобится (и это важно!)

Дорогие и сложные редакторы вроде Adobe Dreamweaver или Microsoft Visual Studio Code (ПОКА не понадобится!). Они мощные, но перегружены функциями, которые на старте только мешают и пугают.

Платное программное обеспечение. Всё, что нам нужно — абсолютно бесплатно.

² Скачивать здесь: <https://notepad-plus-plus.org/>

Интернет-соединение для написания кода. Оно понадобится только позже, чтобы посмотреть справочник или загрузить готовый сайт в сеть. Писать HTML-код и проверять его в браузере можно полностью оффлайн.

4. Практическая подготовка: Давайте проверим прямо сейчас!

Найдите Блокнот: Откройте меню "Пуск" (значок Windows в левом нижнем углу) и в поисковой строке введите "Блокнот". Нажмите на него.

Найдите свой браузер: Скорее всего, его значок (буква "Ch" для Chrome, синяя буква "e" для Edge и т.д.) уже есть на Вашем Рабочем столе или в нижней панели задач.

Создайте папку: На Рабочем столе щелкните правой кнопкой мыши -> "Создать" -> "Папку". Назовите ее "МойСайт". Это будет наша рабочая директория, где мы будем хранить все файлы.

Вот и всё! Ваша "студия веб-разработки" готова к работе.

Заключение:

Использование простейших инструментов — это не ограничение, а осознанная стратегия. Она позволяет нам сосредоточиться на главном — на изучении сути HTML, а не на освоении сложных программ. Помните, великие писатели создавали шедевры с помощью простого пера и бумаги. А наш "Блокнот" — это и есть наше перо, с помощью которого мы будем "писать" наш первый сайт.

Как построены уроки: теория → пример → практическое задание

1. Введение: Почему именно такая структура?

Представьте, что Вы учитесь готовить новое блюдо. Вам не просто дадут список продуктов (теория) и скажут: "Жарь!". Скорее всего, Вы сначала прочитаете рецепт (теория), потом посмотрите, как это делает шеф-повар на видео (пример), и только потом повторите все действия на своей кухне (практика).

Наши уроки построены по точно такому же, самому эффективному принципу. Это проверенная временем педагогическая модель, которая гарантирует не просто "ознакомление с темой", а реальное усвоение материала и выработку навыка.

2. Детальный разбор каждого этапа

Этап 1: ТЕОРИЯ (Объясняем "Что?" и "Зачем?")

Что это такое? Это компактная порция знаний, которая отвечает на два ключевых вопроса:

Что это? (Даем четкое определение)

Для чего это нужно? (Объясняем практическую пользу)

Как это подается?

Минимум терминов: Мы будем использовать специальные термины (например, "тег", "атрибут"), но каждый раз сразу же давать им простое и понятное объяснение через аналогии.

Акцент на логике: Мы не будем заставлять Вас заучивать информацию. Мы будем объяснять логику, чтобы Вы понимали, почему элемент работает именно так.

Конкретика: Вместо "Сегодня мы изучим базовые элементы текста" мы скажем: "Сегодня мы научимся делать заголовки и абзацы, чтобы Ваш текст был аккуратным и читаемым".

Пример из урока:

Тема: Тег абзаца <p>.

Теория: "Тег <p> — это как красная строка в книге. Он сообщает браузеру: "Всё, что находится между открывающим тегом <p> и

закрывающим </p>, является одним абзацем. Сделай отступ снизу и сверху". Это нужно, чтобы текст на странице не был сплошной "простыней", а был удобочитаемым".

Этап 2: ПРИМЕР (Показываем "Как?")

Что это такое? Это наглядная демонстрация того, как только что изученная теория воплощается в реальном коде.

Как это подается?

"Голая" схема: Сначала мы покажем чистый код, чтобы Вы увидели структуру.

Построчный разбор: Мы будем комментировать каждую строчку кода, повторяя и закрепляя теоретические выкладки.

Визуализация результата: Рядом с кодом мы всегда покажем, как этот код будет выглядеть в браузере. Это формирует прямую связь между действием и результатом.

Продолжение примера с тегом <p>:

Пишем в Блокноте:

html

```
<p>Это мой первый абзац. Здесь я расскажу немного о себе.</p>
<p>А это уже второй абзац. В нем я могу написать о своих
увлечениях.</p>
```

Показываем в браузере:

Это мой первый абзац. Здесь я расскажу немного о себе.
А это уже второй абзац. В нем я могу написать о своих увлечениях.

Комментарий: "Видите? Два тега <p> создали два независимых абзаца с отступом между ними".

Этап 3: ПРАКТИЧЕСКОЕ ЗАДАНИЕ (Делаем сами!)

Что это такое? Это ваше задание, где Вы применяете полученные знания на практике. Это самый важный этап урока.

Как это построено?

Четкая формулировка: Задание не будет звучать как "Попрактикуйтесь". Оно будет конкретным: "Добавьте на Вашу страницу index.html три абзаца текста, используя тег <p>".

Пошаговая инструкция (если нужно): Для более сложных заданий мы дадим алгоритм: "1. Откройте Ваш файл в Блокноте. 2. Найдите место после заголовка. 3. Добавьте там первый тег <p>..."

От простого к сложному: Сначала Вы просто повторите пример. Потом — добавите что-то свое.

Вариативность: Часто мы будем давать задание на выбор, чтобы Вы могли проявить творчество. Например: "Напишите абзац о своем домашнем питомце ИЛИ о Вашем любимом времени года".

Завершение примера с тегом <p>:

Ваше задание: "Откройте Вашу страницу index.html. Под главным заголовком <h1>, который мы сделали ранее, добавьте два абзаца <p>. В первом напишите, где Вы живете. Во втором — какое у Вас хобби".

Ожидаемый результат: После выполнения у Вас будет реальный, работающий файл, который стал более содержательным и структурированным.

3. Что делать, если на каком-то этапе возникли трудности?

Наша структура — это Ваша дорожная карта. Если Вы споткнулись, всегда можно вернуться на шаг назад:

Не получается выполнить практическое задание? → Вернитесь к ПРИМЕРУ. Перепишите код оттуда, чтобы "почувствовать" его.

Не понимаете пример? → Перечитайте ТЕОРИЮ. Возможно, Вы упустили ключевую мысль.

Заключение:

Эта трехступенчатая система — "**Объяснили → Показали → Сделали**" — создает прочные нейронные связи в мозге. Она превращает абстрактное знание в реальный навык. В конце каждого урока у Вас на руках будет не просто "пройденная тема в учебнике", а осозаемый результат — Ваша веб-страница, которая стала немного лучше, сложнее и интереснее. Это и есть самый главный источник мотивации для продолжения обучения.

Часть 1:

Первые шаги. Знакомство с миром веб-страниц

Глава 1:

Введение в мир Интернета и веб-страниц

1.1. Что такое Интернет и Всемирная паутина (простыми словами)

Введение: Давайте сначала разделим два понятия

Очень часто слова "*Интернет*" и "*Всемирная паутина*" (или просто "*паутина*") используют как одно и то же. Но на самом деле, это два разных, хоть и тесно связанных, понятия. Давайте разберемся с помощью простой аналогии.

Аналогия с дорогами и магазинами

Представьте себе:

Интернет — это огромная сеть дорог, улиц и магистралей в гигантской стране.

Всемирная паутина (World Wide Web) — это магазины, библиотеки, кинотеатры и кафе, которые расположены вдоль этих дорог.

Теперь давайте разложим эту аналогию по полочкам.

Часть 1: Что такое Интернет? (Дороги и транспорт)

Интернет — это мировая сеть соединенных между собой компьютеров.

Это "Сеть сетей".

Представьте, что компьютер в Вашей квартире — это Ваш личный дом. Ваш дом подключен к улице (ваш провайдер — например, Ростелеком, Билайн).

Ваша улица соединяется с другими улицами, районами и городами. В итоге, все компьютеры, смартфоны, серверы и умные устройства по всему миру соединены в одну гигантскую "*паутину*" проводов (оптоволокно, телефонные линии) и беспроводных каналов (Wi-Fi, спутники). Это и есть Интернет.

Для чего нужен Интернет?

Его главная задача — **пересылать информацию** из одного места в другое, как грузовики перевозят товары по дорогам.

Эта информация делится на маленькие кусочки — "пакеты" — и путешествует по сети самыми быстрыми маршрутами.

Что "ездит" по этим дорогам?

Не только веб-страницы! По "дорогам" Интернета ездят:

Электронные письма (как почтовые машины).

Сообщения в мессенджерах (Viber, WhatsApp) — как быстрые курьеры.

Видеозвонки (Skype, Zoom) — как живое общение по видеосвязи.

Онлайн-радио и музыка — как потоковые трансляции.

Вывод по Части 1: Интернет — это инфраструктура, "сосудистая система", которая обеспечивает передачу любых цифровых данных.

Часть 2: Что такое Всемирная паутина (World Wide Web)? (Магазины и витрины)

Всемирная паутина (WWW) — это огромное собрание документов, сайтов и страниц, которые хранятся на компьютерах, подключенных к Интернету, и связаны между собой ссылками.

Это "Мир информации в одном месте".

Возвращаемся к нашей аналогии: если Интернет — это дороги, то Всемирная паутина — это все те **магазины, библиотеки и кинотеатры** (сайты), в которые Вы заходите, чтобы что-то купить, почитать или посмотреть.

Когда Вы открываете браузер и заходите на сайт новостей, в социальную сеть или на этот учебник — Вы пользуетесь именно Всемирной паутиной.

Как она устроена?

Веб-страница — это один документ, как одна страница в журнале или одна витрина в магазине.

Веб-сайт — это набор таких страниц, объединенных общей темой, как целый магазин с разными отделами.

Ссылки (гиперссылки) — это "двери" и "указатели" между страницами и сайтами. Вы кликаете на ссылку — и "переходите" в другой отдел магазина или даже в другой магазин через дорогу. Именно эти связи-ссылки и создают "паутину".

Ключевая роль браузера.

Браузер (Chrome, Edge и т.д.) — это ваш личный автомобиль с шофером.

Вы говорите ему адрес (например, yandex.ru), и он "едет" по дорогам Интернета, находит нужный "магазин" (сервер, где хранится сайт), забирает оттуда "товар" (HTML-код, картинки) и привозит его Вам, красиво разложив на Вашем экране.

Часть 3: Сводная таблица: Интернет vs. Всемирная паутина

Особенность	Интернет (Дороги)	Всемирная Паутина (WWW) (Магазины)
Суть	Физическая сеть соединенных компьютеров	Информационное пространство, построенное поверх Интернета
Основная функция	Передача данных	Предоставление информации в виде связанных документов (сайтов)
Что использует	Провода, серверы, спутники	Браузеры, веб-сайты, HTML-страницы
Примеры	Электронная почта, видеозвонки, файлообмен	Сайты Google, Википедия, соцсети, онлайн-магазины
Аналогия	Дороги, тунNELи, мосты	Магазины, библиотеки, афиши, витрины на этих дорогах

Заключение: Простой итог

Интернет — это "КАК". Это способ соединения, техническая основа, которая позволяет данным путешествовать.

Всемирная паутина — это "ЧТО". Это то, что мы видим и с чем взаимодействуем с помощью браузера: сайты, страницы, ссылки.

И самое главное для нас:

Язык **HTML** — это тот самый язык, на котором написаны все "вывески", "афиши" и "витрины" во Всемирной паутине. Изучая HTML, мы учимся создавать свои собственные "магазинчики" в этой гигантской и удивительной цифровой стране.

1.2. Как браузер (Chrome, Edge, Яндекс) показывает нам страницы?

Введение: Браузер — наш личный переводчик и дизайнер

Представьте, что Вы получили письмо на незнакомом языке. Чтобы его прочитать, Вам нужен переводчик. А если к письму приложен чертеж будущего дома, Вам нужен архитектор, чтобы превратить этот чертеж в красивую 3D-модель.

Браузер — это и переводчик, и архитектор одновременно. Его работа состоит из нескольких этапов, которые происходят за доли секунды.

Аналогия: Ресторан

Давайте представим весь процесс на простом и вкусном примере:

Вы (клиент) — хотите поесть (получить информацию).

Браузер (ваш персональный официант и шеф-повар) — выполняет Ваш заказ.

Адресная строка (меню с названием блюда) — куда Вы вводите название того, что хотите (адрес сайта).

Сервер (кухня ресторана) — где хранятся все продукты и рецепты (файлы сайта).

HTML-код (рецепт блюда) — инструкция для повара, что и в каком порядке делать.

Готовая страница (готовое блюдо, красиво сервированное на тарелке) — то, что Вы в итоге видите на своем столе (экране).

Этап 1: Вы делаете "заказ"

Что Вы делаете: Вы открываете браузер (Chrome, Edge и т.д.) и вводите в адресную строку название сайта (например, dzen.ru) или полный адрес.

Что делает браузер: Ваш "официант" (браузер) берет этот заказ и отправляет его на "кухню" (сервер), где хранится сайт. Это называется "**отправка запроса**".

Важно: На этом этапе Вы просто попросили принести Вам блюдо под названием "**главная страница dzen.ru**".

Этап 2: "Кухня" (сервер) готовит "ингредиенты"

Что происходит на сервере: "Кухня" получает Ваш заказ. Повар находит нужный "рецепт" — в нашем случае это **HTML-файл**. Этот файл — не готовое блюдо, а именно инструкция.

Что отправляется назад: Сервер ("кухня") отправляет этот HTML-файл (рецепт) обратно вашему браузеру ("официанту").

| **Например, HTML-файл может выглядеть так (это наш "рецепт"):**

```
|   html
<h1>Последние новости</h1>
<p>Сегодня солнечная погода.</p>

```

Этап 3: "Шеф-повар" (браузер) читает рецепт и готовит блюдо

Это самый главный этап, который мы и будем изучать! Браузер получает "рецепт" (HTML-код) и начинает его "готовить" (визуализировать). Этот процесс называется "**рендеринг**".

Давайте посмотрим, как браузер читает каждую строчку нашего "рецепта":

```
<h1>Последние новости</h1>
```

Браузер читает: "Ага! Тег **<h1>**. Это значит — **самый главный заголовок на странице**".

Браузер делает: Он берет текст "**Последние новости**" и отображает его на экране **крупным, жирным шрифтом**.

```
<p>Сегодня солнечная погода.</p>
```

Браузер читает: "Тег **<p>**. Это **обычный абзац текста**".

Браузер делает: Он выводит текст "**Сегодня солнечная погода.**" **обычным шрифтом** и делает отступ от заголовка и от следующего элемента.

```

```

Браузер читает: "Тег ****. Значит, тут нужна картинка. Атрибут **src="sun.jpg"** говорит, что файл с изображением называется **sun.jpg**".

Браузер делает: Он снова отправляет "официанта" на "кухню" (сервер), чтобы тот принес файл с картинкой sun.jpg. Когда картинка приходит, браузер вставляет ее в указанное место.

Этап 4: "Подача блюда" (показ готовой страницы)

После того как браузер прочитал ВЕСЬ HTML-код, загрузил все картинки и применил все стили, он собирает все это в единое целое.

Результат — это **готовая, красиво сверстанная веб-страница**, которую Вы видите на своем экране.

Вы можете легко проверить, как выглядел исходный "рецепт". **Нажмите правой кнопкой мыши на любой странице и выберите**

"Просмотреть код" (или **"Исследовать элемент"**). Откроется окно, где Вы увидите тот самый HTML-код, который браузер получил и обработал. Не бойтесь, Вы ничего не сломаете, просто закройте это окно.

Итог: Краткая схема работы браузера

ЗАПРОС: Вы → Браузеру: "Хочу вот этот сайт".

ПОИСК: Браузер → Серверу: "Дай мне файлы для этого сайта".

ОТВЕТ: Сервер → Браузеру: "Вот HTML-код (и другие файлы)".

ПОСТРОЕНИЕ: Браузер читает код и, как конструктор, собирает из тегов (**<h1>**, **<p>**, ****) готовую страницу.

ПОКАЗ: Вы видите на экране не код, а готовый, понятный документ.

Почему это важно для нас?

Потому что, изучая HTML, мы учимся писать те самые "рецепты" для браузера. Мы будем говорить с ним на понятном ему языке разметки, чтобы он мог превращать наши текстовые инструкции в красивые и понятные страницы. **Вы становитесь тем, кто пишет рецепт для шеф-повара!**

1.3. Что такое HTML? Не язык программиста, а "язык разметки" — как инструкция для браузера.

Введение: Давайте сразу уберем главное заблуждение

Часто, когда говорят "он занимается сайтами", думают, что это "программист". Но создание обычной веб-страницы — это не всегда программирование. HTML — это не язык программирования, и это прекрасная новость! Почему? Потому что его намного легче выучить. Давайте разберемся с помощью простых аналогий.

Часть 1: Язык программирования vs³ Язык разметки

Язык программирования (например, Python, JavaScript) — это набор команд для выполнения действий и принятия решений. Он говорит компьютеру: "*Если пользователь нажал на кнопку, посчитай сумму этих чисел и покажи результат. Если сумма больше 100, покажи скидку*".

Аналогия: Это инструкция для повара с сложными шагами: "*Взвесь 200 грамм муки, взбей 2 яйца, если тесто густое, добавь молока, иначе поставь в духовку на 30 минут*". Здесь есть логика и условия.

Язык разметки (HTML) — это набор специальных пометок (тегов) для описания структуры и содержимого документа. Он говорит браузеру: "Вот это — заголовок, вот это — абзац, а вот здесь — картинка". Он не производит вычислений и не принимает решений.

Аналогия: Это разметка для верстальщика в издательстве. Вы берете обычный текст и помечаете его: "[ЗАГОЛОВОК]Моя дача[/ЗАГОЛОВОК]. [АБЗАЦ]В прошлые выходные я посадил помидоры...[/АБЗАЦ]. [КАРТИНКА]photo1.jpg [/КАРТИНКА]". Вы не говорите, как именно верстать, Вы просто обозначаете, что есть что.

Вывод: Изучая HTML, мы не учимся "программировать", мы учимся правильно структурировать и размечать текст, чтобы браузер понял, как его показывать.

Часть 2: HTML — это "Скелет" сайта

Представьте, что веб-сайт — это человек:

HTML — это скелет и внутренние органы. Он определяет, где будет голова (заголовок), где будут руки и ноги (навигация, основной контент),

³ vs — сокращение: "против"

где сердце (главный текст). Без скелета человек не сможет стоять. Без HTML нет структуры страницы.

CSS (язык стилей, который мы упомянем позже) – это **кожа, волосы, одежда и макияж**. Он отвечает за красоту: цвет фона, размер шрифта, расположение блоков. Он "надевается" поверх HTML-скелета.

JavaScript (язык программирования) – это **мышцы и мозг**. Он отвечает за взаимодействие: анимации, реакцию на нажатие кнопок, обновление информации без перезагрузки страницы.

Таким образом, HTML — это фундамент. Все начинается с него.

Часть 3: Как работает HTML? Знакомимся с "Тегами"

Основа HTML — это **теги**. Тег — это специальная пометка в угловых скобках.

Примеры тегов:

<h1> — заголовок самого высокого уровня (самый крупный).

<p> — абзац.

 — картинка.

<a> — ссылка.

Какими пользоваться? Очень просто! Большинство тегов — парные.

Открывающий тег: <название_тега> — сообщает браузеру: "Внимание! С этого момента начинается заголовок (или абзац, или список)".

Содержимое: Текст, который Вы хотите пометить.

Закрывающий тег: </название_тега> — сообщает браузеру: "Здесь заголовок (или абзац, или список) заканчивается".

Разберем на живом примере:

Что Вы пишете в HTML-файле (Ваша "инструкция"):

```
html
<h1>Мой любимый рецепт</h1>
<p>Это рецепт моих знаменитых драников.</p>
```

Что видит браузер, когда читает эту "инструкцию":

"Ага! Тег **<h1>**. Значит, всё, что до тега **</h1>**, — это главный заголовок. Сделаю его большим и жирным".

"Тег **<p>**. Значит, следующий текст — это абзац. Сделаю отступы сверху и снизу".

Как это увидит пользователь на странице:

```
<h1 style="font-size: 2em; font-weight: bold; margin: 0.67em 0;">Мой любимый рецепт</h1> <p style="margin: 1em 0;">Это рецепт моих знаменитых драников.</p>
```

Часть 4: Итог. Краткий словарик первых тегов

Давайте соберем все аналогии воедино и запомним самое главное.

Понятие	Простое объяснение	Бытовой пример
HTML	Язык разметки. Инструкция по сборке "скелета" страницы.	Чертеж дома. На нем обозначены несущие стены, окна, двери, но не цвет обоев.
Тег	Пометка-ярлык в угловых скобках.	Ценник или этикетка в магазине. Написано "[МОЛОКО]" — значит, это молоко.
Структура	Порядок расположения заголовков, абзацев, картинок на странице.	Оглавление в книге. Где предисловие, где глава 1, где заключение.

Заключение:

HTML — это Ваш способ общения с браузером. Вы не пишете программу, Вы даете ему четкие и простые указания: "Здесь — заголовок, здесь — абзац, а тут вставь картинку". Браузер — очень старательный и послушный работник. Он сделает все именно так, как Вы его попросите. Ваша задача — научиться правильно его просить, используя нужные теги.

Именно этим мы с Вами и займемся в следующих главах. Это увлекательно и совсем не страшно!

1.4. Практика: открываем браузер и смотрим на любую страницу "глазами компьютера" (используем инструмент "Просмотр кода").

Введение: Первый шаг в мир веб-разработки

Сейчас мы с Вами совершим небольшое, но очень важное волшебство. Мы заглянем "за кулисы" любой веб-страницы и увидим то, что обычно скрыто от обычных пользователей — её исходный код. Это тот самый HTML, о котором мы только что говорили.

Не бойтесь! Это абсолютно безопасно. Вы ничего не сломаете и не измените на самом сайте. Вы просто смотрите, как дизайнер смотрит на чертеж здания. Вы — исследователь.

Часть 1: Подготовка — Выбор "объекта" для изучения

Давайте выберем для начала простую и знакомую страницу.

Откройте Ваш браузер (Google Chrome, Яндекс.Браузер, Microsoft Edge или любой другой).

Перейдите на главную страницу одной из этих поисковых систем:

yandex.ru
google.ru
rambler.ru

Мы выбираем их, потому что их страницы хорошо сделаны, и мы сразу увидим много интересного.

Часть 2: Инструкция по открытию "закулисья"

Сделать это можно двумя основными способами. Выберите тот, который Вам удобнее.

Способ 1: Простой и универсальный (с помощью правой кнопки мыши)

Откройте сайт (например, Яндекс).

Наведите курсор мыши на любое свободное место на странице (например, рядом с логотипом или на белом фоне).

Нажмите правую кнопку мыши. Откроется меню.

В этом меню найдите пункт и нажмите на него:

В Google Chrome или Яндекс.Браузере: Просмотреть код или Исследовать элемент.

В Microsoft Edge: Проверить элемент или Прочитать исходный код.

Совет: Если у Вас браузер на английском языке, ищите пункт Inspect или View page source.

Способ 2: С помощью клавиатуры (быстрый способ)

Откройте сайт.

Нажмите на клавиатуре всего одну клавишу: F12. Это самая быстрая и профессиональная клавиша для этого действия.

В большинстве браузеров этого достаточно, чтобы панель инструментов открылась.

Часть 3: Что Вы увидите и как с этим быть?

После того как Вы выполните одно из этих действий, сбоку или внизу экрана появится новое окно или панель с непонятным на первый взгляд кодом. Не пугайтесь! Давайте разберемся, что это.

Вы увидите примерно такой текст (это пример для Яндекса):

```
html
<!DOCTYPE html>
<html>
<head>
<title>Яндекс</title>
<meta charset="utf-8">
</head>
<body>
<div class="content">

<form action="/search">
<input type="text" name="text">
<button type="submit">Найти</button>
</form>
</div>
</body>
</html>
```

Ваши действия — практическое исследование:

Найдите знакомые теги.

Найдите тег <title>. Как Вы думаете, что он означает? (Подсказка: посмотрите на название вкладки в Вашем браузере).

Найдите теги . Это картинки, например, логотип.

Найдите текст, который Вы видите на самой странице (например, слово "Найти" на кнопке). Вы увидите, что в коде он окружен какими-то тегами.

Используйте "волшебный" инструмент — выделение.

В открывшейся панели (если Вы выбрали "Просмотреть код" / Inspect) наведите курсор мыши на разные строки кода.

Вы увидите, что на самой странице браузер будет подсвечивать цветным контуром тот элемент, которому соответствует эта строка кода. Это как рентгеновский аппарат, который показывает скелет сайта!

Попробуйте: Наведите курсор на строку с . Вы увидите, как на странице подсветится логотип. Наведите на строку с <button>..., и подсветится кнопка.

Посмотрите на структуру.

Обратите внимание на слова <head> (голова) и <body> (тело). Всё, что Вы видите на странице (логотипы, кнопки, текст), находится внутри тега <body>.

Часть 4: Что делать, если страница выглядит очень сложной?

Некоторые сайты (например, социальные сети) имеют очень сложный код. Если Вы открыли vk.com или ok.ru и у Вас в панели браузера "поехала крыша" от обилия кода — это нормально!

Вот что можно сделать в этом случае:

Вернитесь на простую страницу (Яндекс или Google).

Используйте инструмент выделения (п.2 из предыдущей части). Это самый наглядный способ понять связь между кодом и тем, что Вы видите.

Не пытайтесь понять ВЕСЬ код сразу. Ваша задача на этом этапе — просто увидеть, что у каждой кнопки и каждого заголовка "под капотом" есть свой HTML-тег. Как у машины есть свой двигатель, который Вы можете посмотреть, открыв капот.

Часть 5: Заключительный шаг — как закрыть панель

Чтобы вернуться к обычному виду страницы, просто:

Снова нажмите клавишу F12.

Или найдите и нажмите на **крестик (X)** в правом верхнем углу открывшейся панели с кодом.

Итог этого занятия:

Поздравляю! Вы только что совершили свой первый шаг в мир веб-разработки. Вы не просто услышали теорию, а своими глазами увидели, что:

Любая страница в интернете состоит из HTML-кода.

Код — это не хаос, а структура из тегов.

Браузер — это программа, которая превращает этот код в удобную для нас страницу.

Запомните это чувство "исследователя". В следующих уроках мы с Вами не будем просто смотреть на этот код, а начнем писать его сами в нашем Блокноте.

Глава 2:

Подготовка рабочего места

2.1. Создаём свою рабочую папку на компьютере "Мой первый сайт".

Введение: Порядок на столе — порядок в голове

Любое дело начинается с организации рабочего места. Если Вы шьёте, Вы раскладываете нитки, ткани и ножницы на столе. Если готовите — подготавливаете продукты на кухне. Создание сайта — не исключение. Нам нужно создать специальную папку, где будут храниться ВСЕ файлы нашего будущего сайта: HTML-страницы, картинки, позже — стили. Это наш "*цифровой рабочий стол*". Сделаем это один раз, и это избавит нас от хаоса и потери файлов в будущем.

Часть 1: Зачем это нужно?

Концентрация: Все файлы проекта будут в одном месте. Не нужно искать их по всему компьютеру.

Порядок: Вы точно будете знать, где что лежит. Это особенно важно, когда файлов станет много.

Корректная работа сайта: Браузер ищет картинки и другие страницы по указанным в коде путям. Если все файлы лежат в одной папке, пути будут простыми, и всё будет работать правильно.

Простота публикации: Когда мы захотим показать наш сайт миру, нам нужно будет просто скопировать одну эту папку на хостинг.

Часть 2: Пошаговая инструкция по созданию папки

Мы будем создавать папку на **Рабочем столе**, чтобы она всегда была у нас перед глазами. Это самый удобный и наглядный способ.

Шаг 1: Откройте Рабочий стол

Закройте или сверните все открытые окна, чтобы увидеть свой **Рабочий стол**. Это главный экран Вашего компьютера, на котором находятся значки программ, файлов и корзины.

Шаг 2: Вызовите меню создания

Наведите курсор мыши на любое свободное место на Рабочем столе (туда, где нет никаких значков).

Нажмите правую кнопку мыши. Откроется контекстное меню — список действий, которые можно выполнить.

Шаг 3: Создайте новую папку

В открывшемся меню наведите курсор на пункт "**Создать**". Рядом появится дополнительное меню.

В этом дополнительном меню выберите самый первый пункт — "**Папку**".

Если у Вас английская версия Windows: ищите пункты New -> Folder.

Шаг 4: Дайте папке понятное имя

Сразу после создания папки её название будет подсвеченено синим цветом, и Вы сможете его редактировать.

Сотрите стандартное название "*Новая папка*".

Ведите новое имя: Мой первый сайт

Нажмите клавишу Enter на клавиатуре, чтобы сохранить название.

Важно! Название папки может быть любым, но следуйте простым правилам:

Не используйте специальные символы: \ / : * ? " < > |

Лучше используйте только русские или английские буквы, цифры, пробелы и дефисы.

Часть 3: Что делать, если не получается?

Ситуация 1: Я случайно нажал не туда и меню закрылось.

Решение: Ничего страшного! Просто снова нажмите правой кнопкой мыши на пустом месте Рабочего стола и повторите шаги.

Ситуация 2: Я не успел дать имя, и папка сохранилась как "*Новая папка*".

Решение: Это легко исправить. Есть такой способ:

Наведите на папку курсор, нажмите правую кнопку мыши и выберите пункт "*Переименовать*".

Ситуация 3: Я не могу найти Рабочий стол, всё закрыто окнами.

Решение: В самом низу экрана есть длинная серая полоса — это "**Панель задач**". На ней обычно есть кнопка "Пуск" (значок Windows) и значки программ. Справа на этой панели есть маленький вертикальный прямоугольник. **Нажмите на него** — это кнопка "**Свернуть все окна**". Все окна исчезнут, и Вы увидите Рабочий стол.

Часть 4: Практическое задание — закрепим результат

Чтобы убедиться, что всё сделано правильно, выполните следующие действия:

Откройте Вашу новую папку. Для этого просто **дважды щелкните по ней левой кнопкой мыши**. Откроется пустое окно. Это и есть внутреннее пространство нашей папки. Сейчас она пуста.

Создайте внутри неё ещё одну папку с названием картинки. Это поможет нам с самого начала приучить себя к порядку. Все фотографии и изображения для сайта мы будем складывать сюда. Чтобы сделать это, находясь внутри папки "*Мой первый сайт*", снова **нажмите правую кнопку мыши** → **Создать** → **Папку** и назовите её "*images*" (картинки).

В итоге у Вас должна получиться такая структура:

text

Рабочий стол
└─ Мой первый сайт
 └─ images

Заключение

Поздравляю! Вы только что создали фундамент для своего будущего сайта. Эта папка — Ваш личный "*творческий цех*". Всё, что мы будем делать дальше — создавать HTML-файлы, добавлять картинки — мы будем сохранять именно здесь.

Запомните: Хорошо организованное рабочее пространство — это 50% успеха в любом деле. Теперь Вы полностью готовы к тому, чтобы приступить к самому интересному — созданию своей первой веб-страницы.

2.2. Наш главный инструмент — Блокнот (или его улучшенная версия Notepad++). Почему не Word?

Введение: Выбираем правильный инструмент для задачи

Представьте, что Вам нужно забить гвоздь. Вы можете сделать это молотком, а можете — дорогой хрустальной вазой. И то, и другое — тяжёлые предметы, но ваза для этого не предназначена: она может разбиться и не дать нужного результата.

Так же и с созданием веб-страниц. Нужен правильный, "профессиональный" инструмент. И для HTML таким идеальным инструментом является простой текстовый редактор, а не текстовый процессор вроде Word.

Часть 1: Почему НЕЛЬЗЯ использовать Microsoft Word?

Word — прекрасная программа для своих задач: для написания писем, рефератов, докладов. Но для программирования и вёрстки он категорически не подходит. Вот главные причины:

1. Word добавляет невидимый мусор.

Что происходит: Когда Вы сохраняете файл в Word, он добавляет в него массу служебной информации: данные о шрифтах, размерах полей, стилях абзацев, настройках документа. Это нужно, чтобы Ваш документ красиво выглядел при печати или открытии в Word у другого человека.

Результат для HTML: Браузер, читая этот "грязный" код, перестаёт понимать, где же тут сами HTML-теги. Он видит кучу лишних команд и символов, не понимает их и показывает пустую страницу или совсем не то, что Вы ожидали.

Простая аналогия: Это как если бы Вы написали чистое письмо другу, а почта добавила бы поверх Вашего текста свои штампы, рекламные листовки и служебные пометки невидимыми чернилами. Друг, получив такое письмо, с трудом смог бы разобрать Ваш исходный текст.

2. Word пытается быть "умным" и всё исправлять.

Пример: Вы пишете угловую скобку <, а Word думает, что это опечатка, и может автоматически заменить её на кавычку или что-то ещё. Он может автоматически сделать первую букву предложения заглавной, а в коде это ломает команды.

Результат для HTML: Ваш чистый, правильный код оказывается искажённым. Браузер показывает ошибку, а Вы не можете понять, в чём дело, ведь в Word всё выглядело нормально.

3. Проблема с сохранением.

Чтобы сохранить файл из Word как веб-страницу, нужно использовать специальный пункт "Сохранить как веб-страницу". Но даже в этом случае Word создаёт очень сложный, раздутый и не оптимальный код, который трудно читать и редактировать.

Вывод: Word создан для визуального оформления документов для людей. Браузеру же нужен чистый, структурированный код, понятный машине.

Часть 2: Почему Блокнот (Notepad) — это идеально?

Обычный Блокнот, который есть в любой версии Windows, — это простейший текстовый редактор. И в этом его главная сила!

1. Он сохраняет только чистый текст.

Что это значит: Блокнот не добавляет в Ваш файл НИКАКОЙ лишней информации. Ни форматирования, ни стилей, ни служебных данных. Только те символы и буквы, которые Вы ввели с клавиатуры.

Результат для HTML: Браузер получает именно тот HTML-код, который Вы написали, без примесей. Если Вы написали <p>Привет</p>, браузер увидит именно <p>Привет</p> и корректно отобразит абзац.

Аналогия: Блокнот — это как чистый лист бумаги и карандаш. Вы пишете то, что думаете. Word — это как набор штампов, трафаретов и автоматических ручек, которые пытаются "улучшить" Ваши слова.

2. В нём нет ничего лишнего.

В Блокноте нет сотен кнопок, меню и функций, которые могут Вас отвлекать и пугать. Вы видите только чистое поле для ввода текста. Это помогает полностью сосредоточиться на изучении синтаксиса HTML.

3. Он есть у всех.

Вам не нужно ничего скачивать и устанавливать. Он уже есть в Вашем компьютере. Это делает старт обучения максимально быстрым и простым.

Недостаток обычного Блокнота: Он слишком прост. В нём нет подсветки синтаксиса, нумерации строк и других удобств для программиста.

Часть 3: Улучшенная альтернатива — Notepad++

Notepad++ — это как Блокнот, но "прокачанный". Это бесплатная программа, которую можно скачать из интернета.

Его главные преимущества:

Подсветка синтаксиса: Он автоматически раскрашивает разные элементы кода в разные цвета.

Теги <html>, <p> — будут синими.

Текст внутри тегов — чёрным.

Значения атрибутов — красным или зелёным.

Зачем это нужно: Цвета помогают глазам быстро находить теги, легче замечать ошибки (например, если Вы забыли закрывающую скобку, цвет всего кода "поедет"). Это не просто красиво, это очень практично.

Нумерация строк: Каждая строка кода имеет свой номер. Если браузер покажет Вам сообщение об ошибке в строке 15, Вы мгновенно найдёте её в редакторе.

Автодополнение: Программа может предлагать Вам варианты тегов, когда Вы начинаете их печатать, что ускоряет работу и уменьшает количество опечаток.

Резюме по выбору инструмента:

Начинайте с обычного Блокнота. Это поможет Вам прочнее усвоить основы, ничего не отвлекая.

Когда освоитесь, установите Notepad++. Он сделает процесс написания кода более удобным и наглядным.

Часть 4: Практическое задание — находим Блокнот

Давайте прямо сейчас найдем и откроем наш главный инструмент.

Способ 1 (через Поиск):

Нажмите кнопку "**Пуск**" (значок Windows в левом нижнем углу). В строке поиска (или в меню "**Все программы**") начните печатать слово "**Блокнот**".

Система найдет программу, нажмите на неё.

Способ 2 (через Стандартные программы):

Нажмите "**Пуск**".

Выберите "**Все приложения**" или "**Все программы**".

Найдите папку "**Стандартные**" и откройте её.

Внутри этой папки найдите и запустите "**Блокнот**".

Заключение:

Помните: могущественные замки строятся с помощью простых и надежных инструментов — молотка и зубила. Не требуются лазеры и сложные аппараты. Так и наш "**Блокнот**" — это простое, но идеально подходящее зубило, с помощью которого мы будем высекать наш первый веб-сайт. Он даёт нам полный контроль над кодом и гарантирует, что браузер поймёт нас с первого раза.

2.3. Создаём первый файл с правильным именем: index.html

Введение: Даём нашему сайту "имя и фамилию"

У каждого документа есть своё название: "Заявление на отпуск", "Рецепт борща", "Письмо Маше". У веб-страниц тоже есть имена, и от имени зависит очень многое. Сегодня мы создадим наш главный файл — index.html. Это "входная дверь" в Ваш сайт, его титульная страница.

Часть 1: Почему именно index.html?

Это не просто случайное имя. Это стандарт, который понимают все браузеры и веб-серверы по всему миру.

"*Index*" — это главная страница.

Аналогия: Представьте, что Вы заходите в магазин. Вы же не просите показать Вам "стенд с хлебом, потом отдел с молоком, потом кассу". Вы просто заходите через главный вход. Файл index.html — это и есть **главный вход** на Ваш сайт.

Когда Вы заходите на сайт (например, yandex.ru), браузер на самом деле ищет и открывает файл с именем index.html (или index.php и т.д.), если Вы не указали другую страницу. Адрес yandex.ru технически равнозначен yandex.ru/index.html.

Расширение .html — это "фамилия" файла.

Расширение (часть имени после точки) сообщает компьютеру, **какого типа этот файл** и какой программой его нужно открывать.

Файл с расширением .txt — текстовый, его открывает Блокнот.

Файл с расширением .jpg — это картинка.

Файл с расширением .html — это **веб-страница**, и его нужно открывать в браузере.

Итог: Назвав файл index.html, мы говорим всем: "Это главная страница веб-сайта, открой её в браузере".

Часть 2: Пошаговая инструкция по созданию файла

Мы создадим файл прямо в папке "Мой первый сайт", которую сделали ранее.

Шаг 1: Откройте Вашу папку

Найдите на Рабочем столе папку "**Мой первый сайт**" и откройте её двойным щелчком мыши. Вы должны увидеть пустое окно (или папку картинки, если Вы её создали).

Шаг 2: Создайте новый файл

Щёлкните правой кнопкой мыши на пустом месте внутри этой папки. В контекстном меню выберите "**Создать**". В списке типов файлов найдите и выберите "**Текстовый документ**".

Шаг 3: Дайте файлу правильное имя

Сразу после создания файла его имя будет выделено синим цветом, и Вы сможете его редактировать. Оно будет называться "*Новый текстовый документ.txt*".

Ведите новое имя: index.html

Внимание! Самое главное: Вы должны **полностью стереть старое имя, включая точку и расширение .txt**, и написать index.html

Нажмите клавишу Enter.

Шаг 4: Подтвердите изменение расширения

Система может показать Вам предупреждение: "*После изменения расширения этот файл может стать недоступным. Вы действительно хотите изменить его?*"

Это абсолютно正常но! Система просто перестраховывается, так как мы меняем тип файла. Смело нажимайте кнопку "**Да**".

Поздравляю! Вы только что создали свою первую веб-страницу! Вы увидите, что значок файла изменился. Скорее всего, он теперь будет выглядеть как значок Вашего браузера (например, хромированный кружок для Chrome). Это значит, что система теперь понимает: этот файл — веб-страница.

Часть 3: Что делать, если не получается?

Проблема 1: У меня не показываются расширения файлов (.txt, .html).

Симптом: Вы переименовали файл в index.html, но он всё равно называется index.html.txt и выглядит как текстовый файл.

Решение: Нужно включить отображение расширений. Сделаем это один раз, и больше проблема не повторится.

Откройте любую папку.

Вверху выберите меню "Вид" (или "View").

Найдите и поставьте галочку напротив пункта "**Расширения имён файлов**".

Проблема 2: Я случайно нажал "Отмена" при переименовании.

Решение: Просто щёлкните по файлу правой кнопкой мыши, выберите "Переименовать" и повторите шаг 3.

Проблема 3: Файл открывается в Блокноте, а не в браузере.

Решение: Чтобы его открыть в браузере, щёлкните по нему **правой кнопкой мыши**, выберите "Открыть с помощью" и выберите Ваш браузер (Chrome, Edge и т.д.). Чтобы он всегда открывался в браузере, в этом же меню можно выбрать "Выбрать другое приложение" и поставить галочку "Всегда использовать это приложение...".

Часть 4: Практическое задание — проверяем результат

Давайте убедимся, что всё работает так, как задумано.

Откройте Ваш файл index.html в браузере.

Самый простой способ: просто **дважды щёлкните по нему левой кнопкой мыши**. Он должен запуститься в Вашем браузере.

Что Вы увидите?

Вы увидите **совершенно пустую белую страницу**. Адресная строка браузера будет показывать путь к вашему файлу, например: C://Users/Имя/Desktop/Мой первый сайт/index.html

Почему страница пустая?

Потому что мы создали файл, но не написали в нём ни строчки кода! Это как купить чистый блокнот — страницы есть, а текста ещё нет.

Это совершенно нормально и именно так и должно быть! Мы создали "чистый холст". В следующем уроке мы начнём "рисовать" на этом холсте с помощью HTML-тегов.

Заключение

Вы только что совершили ключевой технический шаг — создали корректно названный файл веб-страницы. Теперь у Вашего будущего

сайта есть сердце — главная страница index.html. Все дороги Вашего сайта будут вести сюда. Этот файл — основа, на которую мы будем нанизывать весь наш контент: текст, картинки, ссылки.

Запомните этот момент. Сейчас Ваш сайт — это тихий, тёмный и пустой дом. Но в следующих уроках мы включим в нём свет, расставим мебель и повесим картины на стены.

2.4. Практика: создаём папку и файл, открываем его в Блокноте и в браузере.

Введение: Первый полный цикл работы

Сейчас мы с Вами выполним самый важный практический цикл, который станет основой всей нашей дальнейшей работы. Мы проверим нашу "творческую лабораторию" в действии: создадим среду для работы и убедимся, что всё функционирует правильно.

Наша цель на этом занятии: Создать папку Учебный проект, внутри неё — файл practice.html, а затем открыть этот файл в двух программах: в Блокноте (для редактирования) и в браузере (для просмотра результата).

Часть 1: Создаём учебную папку и файл

Это нужно, чтобы мы могли экспериментировать, не боясь что-то испортить в основном проекте.

Шаг 1: Создаём папку на Рабочем столе

Щёлкните правой кнопкой мыши на пустом месте Рабочего стола.

В меню выберите "**Создать**" → "**Папку**".

Дайте папке понятное имя: Учебный проект.

Нажмите Enter.

Шаг 2: Создаём HTML-файл внутри папки

Откройте только что созданную папку Учебный проект (двойной щелчок).

Внутри этой папки щёлкните правой кнопкой мыши по пустому месту.

Выберите "**Создать**" → "**Текстовый документ**".

Сразу же дайте файлу имя: practice.html

Внимание! Важно стереть всё старое имя, включая .txt, и написать новое с расширением .html.

Нажмите Enter. Если система спросит о подтверждении изменения расширения, нажмите "**Да**".

Результат: В папке Учебный проект на Рабочем столе у Вас лежит файл practice.html с иконкой Вашего браузера.

Часть 2: Открываем файл в Блокноте (для редактирования)

Браузер показывает нам результат, а редактировать код мы будем в текстовом редакторе.

Шаг 1: Запускаем Блокнот

Нажмите кнопку "**Пуск**".

Введите в поиске "**Блокнот**" и откройте его.

Шаг 2: Открываем наш файл в Блокноте

В Блокноте в верхнем меню выберите "**Файл**" → "**Открыть...**" (или нажмите Ctrl+O).

В открывшемся окне найдите и перейдите в папку Учебный проект на Рабочем столе.

Важный момент: В правом нижнем углу окна, где написано "Текстовые документы (.txt)", измените этот фильтр на "Все файлы (*.*)". Иначе Вы не увидите свой practice.html, так как он не является .txt-файлом.
Теперь выберите файл practice.html и нажмите кнопку "**Открыть**".

Шаг 3: Добавляем первый тест-код

Вы увидите, что Блокнот открылся, но он абсолютно пустой. Давайте это исправим!

Напишите в Блокноте всего одну строчку:

```
html  
<h1>Ура! У меня получилось!</h1>
```

Теперь сохраните файл: "**Файл**" → "**Сохранить**" (или нажмите Ctrl+S).

Часть 3: Открываем файл в браузере (для просмотра)

Теперь посмотрим, как браузер интерпретирует наш код.

Способ 1: Прямой запуск (самый простой)

Просто вернитесь в папку Учебный проект и **дважды щёлкните левой кнопкой мыши по файлу** practice.html.

Файл автоматически откроется в Вашем браузере по умолчанию (Chrome, Edge и т.д.).

Способ 2: Через меню "Открыть" в браузере

Запустите браузер (например, Chrome).

Нажмите комбинацию клавиш Ctrl+O (буква O, не ноль).

В открывшемся окне найдите Ваш файл practice.html в папке Учебный проект и откройте его.

Что Вы должны увидеть в браузере:

В центре страницы будет крупная надпись: "*Ура! У меня получилось!*". Это браузер отобразил содержимое нашего тега <h1>.

Часть 4: Полный рабочий цикл "Редактирование → Просмотр"

Давайте закрепим самый главный навык — процесс внесения изменений и их немедленной проверки.

Вернитесь в Блокнот с Вашим кодом (<h1>Ура! У меня получилось!</h1>).

Добавьте ещё одну строчку под существующей:

```
html
<h1>Ура! У меня получилось!</h1>
<p>Это мой первый текст на веб-странице.</p>
```

Сохраните файл в Блокноте (Ctrl+S).

Вернитесь в браузер и ОБНОВИТЕ страницу, нажав клавишу F5 или кнопку "Обновить" (круговую стрелку рядом с адресной строкой).

Результат: Под крупным заголовком сразу же появится новый абзац текста: "Это мой первый текст на веб-странице."

Часть 5: Возможные проблемы и их решения

Проблема 1: В браузере виден не текст, а сам код (<h1>...)

Причина: Вы забыли сохранить файл в Блокноте после внесения изменений.

Решение: Вернитесь в Блокнот, нажмите Ctrl+S, затем обновите страницу в браузере.

Проблема 2: Файл открывается в Блокноте, а не в браузере

Причина: Система не ассоциирует .html файлы с браузером.

Решение: Щёлкните по файлу правой кнопкой мыши → "Открыть с помощью" → выберите Ваш браузер. Для постоянного решения в этом

же меню выберите "Выбрать другое приложение", отметьте браузер и поставьте галочку "Всегда использовать...".

Проблема 3: Не видно расширения .html

Решение: Включите отображение расширений файлов через меню "Вид" в любой папке, поставив галочку "Расширения имён файлов".

Заключение и итоги практики

Поздравляем! Вы только что успешно завершили полный цикл веб-разработки в миниатюре:

- Создали структуру проекта (папка Учебный проект).
- Создали веб-страницу (файл practice.html).
- Написали HTML-код в Блокноте.
- Сохранили изменения.
- Просмотрели результат в браузере.
- Внесли правки и увидели их после обновления страницы.

Это именно тот алгоритм, по которому мы будем работать всегда:

Блокнот (редактируем код) → Сохраняем (Ctrl+S) → Браузер (смотрим результат, F5)

Запомните этот ритм. Теперь Вы владеете основным инструментарием и можете самостоятельно создавать и изменять веб-страницы. В следующих уроках мы будем наполнять наши страницы всё более интересным содержимым!

Часть 2: Основы HTML. Пишем первую страницу

Глава 3: Структура каждой HTML-страницы

3.1. Обязательные теги: <html>, <head>, <body>. Кто за что отвечает?

Введение: Строим каркас нашего веб-дома

Прежде чем строить дом, архитектор создаёт чертёж с указанием несущих стен, фундамента и крыши. Так и в HTML есть обязательные теги, которые создают каркас каждой веб-страницы. Это основа, на которую мы будем "навешивать" весь наш контент.

Давайте разберёмся с помощью простой и понятной аналогии.

Часть 1: Аналогия с человеком

Представьте, что Ваша веб-страница — это человек:

Тег <html> — это **всё тело человека** от макушки до пяток. Он является главным контейнером, который заключает в себе всё содержимое страницы.

Тег <head> — это **голова и мозг** человека. В этой части хранится служебная информация, которую не видно на самой странице, но которая крайне важна для её функционирования (название, инструкции для браузера, ссылки на стили).

Тег <body> — это **туловище, руки и ноги**, то есть всё видимое тело. В этой части находится всё, что видят посетители Вашего сайта: текст, картинки, кнопки, видео.

Часть 2: Детальный разбор каждого тега

1. Тег <html> — Главный контейнер

Что это? Это самый внешний тег, который сообщает браузеру: "Всё, что находится между <html> и </html>, — это код HTML-документа".

Зачем нужен? Он задаёт границы документа и помогает браузеру правильно его интерпретировать.

Простая аналогия: Это **конверт** для письма. Всё письмо (и служебные пометки, и сам текст) находится внутри конверта.

2. Тег <head> — Служебная информация (голова)

Что это? Это раздел для метаданных. Данные внутри `<head>` не отображаются на самой странице в виде контента, но они критически важны.

Что обычно хранится в `<head>`?

`<title>` — Заголовок страницы, который отображается на вкладке браузера. Это как **бирка на одежде** — её видно снаружи (на вкладке), но она не является частью самой одежды (тела страницы).

`<meta charset="utf-8">` — Указание кодировки символов. Гарантирует, что русские (и любые другие) буквы будут отображаться правильно, без "кракозябр". Это как **инструкция по чтению** алфавита.

Ссылки на файлы CSS (для оформления) и JavaScript (для интерактивности), которые мы подключим позже.

Простая аналогия: Это **паспорт** человека. Вы не видите его, просто глядя на человека, но в нём содержится вся основная информация: имя (`<title>`), гражданство (кодировка).

3. Тег `<body>` — Видимое содержимое (тело)

Что это? Это основная, видимая часть страницы. Всё, что Вы напишете внутри `<body>`, браузер покажет пользователю.

Что обычно хранится в `<body>`?

Заголовки (`<h1>`, `<h2>`)

Абзацы (`<p>`)

Списки (``, ``)

Картинки (``)

Ссылки (`<a>`)

И всё остальное, что должно быть видно.

Простая аналогия: Это **витрина магазина**. Всё, что Вы размещаете в витрине (`<body>`), видят прохожие (пользователи). А служебные помещения и склад (`<head>`) скрыты от их глаз.

Часть 3: Как это выглядит в коде? Собираем каркас

Давайте соберём все эти теги вместе в нашем файле `index.html`. Это будет наш шаблон, с которого начинается **ЛЮБАЯ** страница.

Откройте Ваш файл `index.html` в Блокноте и напишите следующий код:

`html`

```
<html>
  <head>
    <title>Мой первый сайт</title>
    <meta charset="utf-8">
  </head>
  <body>
    А здесь скоро будет мой первый текст!
  </body>
</html>
```

Теперь давайте разберём его построчно:

<html> — Открывает наш HTML-документ.
<head> — Начинает раздел служебной информации.
<title>Мой первый сайт</title> — Задаёт заголовок вкладки браузера.
<meta charset="utf-8"> — Указывает, что мы используем русские буквы.
</head> — Закрывает раздел служебной информации.
<body> — Начинает видимую часть страницы.
А здесь скоро будет мой первый текст! — Простой текст, который мы увидим на странице.
</body> — Закрывает видимую часть.
</html> — Закрывает весь HTML-документ.

Обратите внимание на отступы. Они не обязательны для браузера, но очень помогают нам, людям, видеть структуру кода и понимать, какой тег где закрывается.

Часть 4: Практическое задание — увидеть разницу

Давайте закрепим материал, сразу же увидев результат.

Сохраните файл в Блокноте (Ctrl + S).

Откройте файл index.html в браузере (или обновите страницу, если он уже открыт).

Посмотрите внимательно и найдите:

Заголовок во вкладке браузера: В самой верхней части окна, на вкладке, Вы должны увидеть текст "Мой первый сайт".

Текст на самой странице: В центре белого окна браузера Вы увидите фразу: "А здесь скоро будет мой первый текст!".

Вывод: Вы наглядно убедились, что содержимое `<head>` (заголовок вкладки) и содержимое `<body>` (текст на странице) отображаются в разных местах и выполняют разные функции.

Заключение

Вы только что освоили фундаментальный принцип структуры HTML-документа. Запомните эту простую схему:

- `<html>` — Всё содержимое.
- `<head>` — То, что не видно, но важно (мозг).
- `<body>` — То, что видно всем (тело).

Этот каркас — основа основ. Любая Ваша будущая веб-страница будет начинаться именно с этих трёх тегов. Теперь наш "дом" имеет прочный фундамент и стены. В следующих уроках мы начнём расставлять в нём "мебель" — добавлять заголовки, абзацы и картинки.

`<!DOCTYPE html>` — это критически важная часть современного HTML-документа, и его отсутствие в оглавлении и объяснении является серьезным упущением.

Давайте исправим это и подробно разберем, что это такое и почему он нужен.

Дополнение к главе 3.1: Волшебная строка `<!DOCTYPE html>`

Введение: Объявляем правила игры

Представьте, что Вы садитесь играть в шахматы. Первое, что Вы делаете — договариваетесь о правилах. `<!DOCTYPE html>` — это и есть наше объявление браузеру: "Мы будем играть по современным и понятным правилам HTML5".

Часть 1: Что такое `<!DOCTYPE>` и зачем он нужен?

`<!DOCTYPE html>` — это **не тег, а декларация (объявление) типа документа**.

Его единственная задача — сообщить браузеру, как именно следует интерпретировать и отображать нашу веб-страницу.

Историческая причина: Проблема "Режимов совместимости"

Раньше, в эпоху "войн браузеров", существовало несколько разных стандартов HTML. Из-за этого старые сайты могли криво отображаться в новых браузерах, и наоборот.

Чтобы решить эту проблему, браузеры стали поддерживать два режима отображения:

Стандартный режим (Standards Mode): Браузер следует современным и единым правилам.

Режим совместимости (Quirks Mode): Браузер пытается имитировать старое, "причудливое" поведение, чтобы "пожертвовать" стандартами ради отображения древних сайтов.

`<!DOCTYPE>` — это переключатель между этими режимами.

ЕСТЬ корректный `<!DOCTYPE>` → браузер включает **Стандартный режим**. Ваша страница отображается предсказуемо и единообразно во всех современных браузерах.

НЕТ <!DOCTYPE> → браузер включает Режим совместимости. Верстка может "поехать", элементы будут отображаться не так, как Вы ожидаете.

Часть 2: Почему именно <!DOCTYPE html> так прост?

В прошлом объявления типа документа (Doctype) были длинными и сложными, так как ссылались на официальные спецификации. Например, для HTML 4.01 он выглядел так:

```
html
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.01 Transitional//EN"
"http://www.w3.org/TR/html4/loose.dtd">
```

Разработчики HTML5 решили эту проблему кардинально. Они максимально упростили декларацию, чтобы её было легко запомнить и написать.

<!DOCTYPE html> — это единственная и достаточная декларация для активации стандартного режима в современном HTML5.

Это не чувствительно к регистру (можно писать <!doctype html>). Это кратко и невозможно забыть.

Часть 3: Где и как его писать?

<!DOCTYPE html> всегда должен быть самой первой строчкой в Вашем HTML-файле. Даже перед тегом <html>.

Правильная, полная структура документа теперь выглядит так:

```
html
<!DOCTYPE html>
<html>
<head>
<title>Мой первый сайт</title>
<meta charset="utf-8">
</head>
<body>
А здесь скоро будет мой первый текст!
</body>
</html>
```

Часть 4: Практическое задание — исправляем наш шаблон

Откройте ваш файл index.html в Блокноте.

Добавьте самую первую строку: <!DOCTYPE html>

Сохраните файл.

Обновите страницу в браузере.

Визуально вы, скорее всего, не заметите разницы. И это хорошо! Это значит, что браузер и так корректно обработал Вашу страницу. Но теперь Вы можете быть на 100% уверены, что он делает это по самым современным и строгим стандартам, а не угадывает, как ему лучше поступить.

Итог и исправление оглавления

Большое спасибо за Вашу внимательность! Это важнейшее уточнение.

Краткий итог:

<!DOCTYPE html> — это первая и обязательная строка любого HTML5-документа.

Назначение: Принудительно включает в браузере современный стандартный режим отображения (Standards Mode).

Местоположение: Всегда строго на первой строке файла.

Исправленное оглавление для Главы 3.1 должно выглядеть так:

Глава 3: Структура каждой HTML-страницы

3.0. Самая первая строка: <!DOCTYPE html> — гарантия правильного отображения.

3.1. Обязательные теги: <html>, <head>, <body>. Кто за что отвечает?

3.2. Заголовок страницы <title>: что мы видим во вкладке браузера.

3.3. Простая аналогия: страница — это человек (<html>). Голова (<head>) — для мыслей (название), тело (<body>) — для содержимого.

3.4. Практика: пишем "скелет" нашей первой страницы.

Ещё раз благодарю Вас. Теперь объяснение структуры HTML-документа является полным и корректным.

3.2. Заголовок страницы <title>: что мы видим во вкладке браузера

Введение: Визитная карточка Вашей страницы

Представьте, что у каждой книги на корешке есть название. Именно по нему Вы находите нужную книгу на полке. Тег <title> — это и есть такое "название на корешке" для Вашей веб-страницы. Это не заголовок внутри статьи, а её официальное "имя" в мире браузера.

Часть 1: Что такое тег <title> и где он находится?

<title> — это парный тег, который находится **внутри раздела <head>** Вашего HTML-документа.

Его содержимое **не отображается на самой веб-странице**, но играет crucial роль.

Правильное расположение в коде:

```
html
<!DOCTYPE html>
<html>
<head>
<title>Вот сюда мы пишем заголовок страницы</title>
<meta charset="utf-8">
</head>
<body>
А это текст, который видят пользователи на самой странице.
</body>
</html>
```

Часть 2: Где мы видим результат действия тега <title>?

Заголовок <title> отображается в трёх ключевых местах:

1. На вкладке браузера

Это самое очевидное место. Заголовок помогает ориентироваться, когда открыто много вкладок.

2. В результатах поисковых систем (Яндекс, Google)

<title> становится заголовком Вашей страницы в поисковой выдаче. От его качества часто зависит, перейдут ли пользователи на Ваш сайт.

3. В закладках браузера

Когда Вы добавляете страницу в закладки, браузер по умолчанию использует содержимое <title> как название закладки.

Часть 3: Чем <title> отличается от заголовка <h1>?

Это частый вопрос новичков. Давайте чётко разграничим:

Характеристика	Тег <title>	Тег <h1>
Местоположение	В разделе <head>	В разделе <body>
Кто его видит?	Браузер, поисковые системы, пользователь (во вкладке)	Непосредственно на странице
Назначение	Идентификация страницы в браузере и поиске	Структурирование содержимого на самой странице
Аналогия	Название книги на корешке	Заголовок первой главы внутри книги

Практический пример:

html

```
<!DOCTYPE html>
<html>
<head>
<title>Рецепт идеальных блинчиков от бабы Глаши</title>
</head>
<body>
<h1>Как испечь самые пышные блинчики</h1>
<p>Сегодня я научу Вас своему фирменному рецепту...</p>
</body>
</html>
```

Во вкладке браузера вы увидите: "Рецепт идеальных блинчиков от бабы Глаши"

На самой странице крупным шрифтом будет: "Как испечь самые пышные блинчики"

Часть 4: Как правильно составлять заголовок <title>?

1. Будьте краткими, но информативными

Плохо: <title>Главная страница</title>

Хорошо: <title>Сад и огород для души - советы садоводам</title>

2. Указывайте ключевую тему страницы

Плохо: <title>Мой блог</title>

Хорошо: <title>Вязание для начинающих: уроки и схемы</title>

3. Добавляйте название сайта (не обязательно для главной)

<title>Выращивание помидоров | Садовод-любитель</title>

4. Оптимальная длина

Старайтесь уложиться в **50-60 символов**, чтобы заголовок полностью отображался в поисковой выдаче.

Часть 5: Практическое задание

Шаг 1: Добавьте заголовок в Ваш файл

Откройте ваш index.html и добавьте или измените тег <title>:

```
html
<!DOCTYPE html>
<html>
<head>
<title>Мой первый сайт - Добро пожаловать!</title>
<meta charset="utf-8">
</head>
<body>
<h1>Здравствуйте! Это моя первая веб-страница</h1>
<p>Я учусь создавать сайты и у меня получается!</p>
</body>
</html>
```

Шаг 2: Посмотрите результат

Сохраните файл (Ctrl + S)

Откройте/обновите страницу в браузере

Найдите три места, где отображается Ваш заголовок:

Вкладка браузера (вверху)

Закладки (добавьте страницу в закладки чтобы проверить)

Поиск (скопируйте текст из <title> и вставьте в поисковой строке)

Шаг 3: Экспериментируйте

Попробуйте изменить текст внутри <title> и посмотрите, как меняется заголовок во вкладке браузера.

Часть 6: Частые ошибки и их последствия

Ошибка 1: Полное отсутствие <title>

Результат: Браузер покажет либо путь к файлу (file:///C:/...), либо просто "Документ"

Как исправить: Всегда добавляйте осмысленный <title>

Ошибка 2: Слишком длинный заголовок

Результат: В поисковой выдаче заголовок обрежется многоточием ...

Как исправить: Сформулируйте мысль короче

Ошибка 3: Однаковые <title> на всех страницах сайта

Результат: Пользователям будет сложно ориентироваться между вкладками

Как исправить: Давайте каждой странице уникальное название

Заключение

Тег <title> — это маленький, но очень важный элемент Вашей веб-страницы. Он:

- Помогает пользователям ориентироваться между вкладками
- Влияет на продвижение в поисковых системах
- Сохраняет информацию о странице в закладках
- Даёт странице официальное "имя"

Запомните: <title> — для браузера и поисковых систем, <h1> — для посетителей Вашего сайта. Теперь Ваша страница имеет не только содержание, но и правильное название!

3.3. Простая аналогия: страница — это человек (<html>). Голова (<head>) — для мыслей (название), тело (<body>) — для содержимого.

Введение: Оживляем код с помощью понятного образа

Изучать абстрактные теги может быть сложно. Гораздо проще, когда у них есть понятный и запоминающийся образ. Давайте "оживим" нашу веб-страницу, представив её в виде человека. Эта аналогия — один из самых эффективных способов раз и навсегда понять структуру HTML.

Часть 1: Человек как целостный организм — тег <html>

Аналогия: Весь человек от макушки до пят.

Что это значит в HTML:

Тег <html> — это оболочка, которая содержит в себе **ВСЁ**, что относится к нашей веб-странице.

Это самый главный, корневой тег. Всё, что не находится внутри него, браузером не учитывается.

Как это выглядит в коде:

```
html
<!DOCTYPE html>
<html>
<!-- ВСЁ, что здесь находится — это наша "личность", наш
целостный документ --&gt;
&lt;/html&gt;</pre>
```

Простая параллель: Когда Вы говорите "*Вот Иван Иванович*", Вы имеете в виду всего человека целиком. Так и <html> — это указание на весь документ целиком.

Часть 2: Голова — тег <head> (невидимая, но важная часть)

Аналогия: Голова и мозг человека.

Что это значит в HTML:

Раздел <head> содержит служебную информацию, мысли, идеи, планы — всё то, что не видно снаружи, но управляет поведением и идентичностью.

Содержимое <head> не отображается на самой странице, но без него страница будет "неполноценной".

Что находится "в голове" (`<head>`) нашей страницы:

`<title>` — Имя человека.

Аналогия: Как Вас зовут? "Меня зовут Анна".

В HTML: Это название страницы, которое видят во вкладке браузера. `<title>Мой дневник</title>`.

`<meta charset="utf-8">` — Язык мышления.

Аналогия: На каком языке Вы думаете? "Я думаю на русском".

В HTML: Это указание браузеру, в какой кодировке читать страницу, чтобы правильно показывать русские буквы.

Будущие элементы: "Знания" и "Навыки".

Аналогия: В голове хранятся знания (как решить задачу) и навыки (как водить машину).

В HTML: Позже мы сюда будем подключать стили CSS (знания о том, как выглядеть красиво) и скрипты JavaScript (навыки для интерактивности).

Как это выглядит в коде:

```
html
<head>
<title>Мой личный дневник</title> <!-- "Имя" -->
<meta charset="utf-8"> <!-- "Язык мышления" -->
<!-- Здесь будут "знания" и "навыки" -->
</head>
```

Часть 3: Тело — тег `<body>` (видимая и активная часть)

Аналогия: Тело, лицо, одежда человека — всё, что видят другие люди.

Что это значит в HTML:

Раздел `<body>` содержит всё, что пользователь видит на странице: текст, картинки, кнопки, ссылки.

Это "витрина" Вашего сайта, его видимое содержимое.

Что находится "в теле" (`<body>`) нашей страницы:

`<h1>` — Крупные черты лица.

Аналогия: "У него добрые глаза и открытая улыбка" — первое, что бросается в глаза.

В HTML: Главный заголовок страницы, который сразу привлекает внимание.

<p> — Речь, слова.

Аналогия: То, что человек говорит.

В HTML: Абзацы текста, основное содержание.

** — Одежда и аксессуары.**

Аналогия: Красивая рубашка, очки, сумка — то, что делает образ завершенным.

В HTML: Картинки, которые делают страницу наглядной и привлекательной.

<a> — Действия и жесты.

Аналогия: Рукопожатие, указание рукой ("проходите сюда").

В HTML: Ссылки, которые ведут на другие страницы или сайты.

Как это выглядит в коде:

```
html
<body>
<h1>Добро пожаловать на мою страничку!</h1> <!-- "Крупные черты" -->
<p>Меня зовут Василий, и я люблю выращивать цветы.</p> <!-- "Речь" -->
 <!-- "Внешний вид" -->
<a href="garden.html">Посмотрите мой сад</a> <!-- "Жест-приглашение" -->
</body>
```

Часть 4: Собираем полную аналогию

Давайте соберём всё вместе:

```
html
<!DOCTYPE html> <!-- (Это как свидетельство о рождении - официальный документ) -->
<html> <!-- (Весь человек: Василий Петров) -->
```

```
<head> <!-- (Голова и мозг Василия) -->
<title>Сад Василия Петрова</title> <!-- (Его имя в паспорте) -->
<meta charset="utf-8"> <!-- (Он думает на русском языке) -->
</head>

<body> <!-- (Тело Василия, которое видят все) -->
<h1>Я выращиваю самые красивые розы</h1> <!-- (Его фирменная улыбка) -->
<p>Заходите в гости полюбоваться моими цветами!</p> <!-- (Его приветственная речь) -->
 <!-- (Его красивая садовая одежда) -->
</body>

</html>
```

Часть 5: Практическое задание "Создайте своего виртуального человека"

Откройте ваш файл index.html в Блокноте

Используя аналогию с человеком, напишите код:

```
html
<!DOCTYPE html>
<html>
<head>
<!-- "Имя" Вашей страницы -->
<title>Страница [Ваше Имя]</title>
<meta charset="utf-8">
</head>
<body>
<!-- "Главная черта" -->
<h1>Всем привет! Я [Ваше Имя]</h1>

<!-- "Что Вы хотите сказать" -->
<p>Я осваиваю создание сайтов и это очень увлекательно!</p>

<!-- "Чем Вы занимаетесь" -->
<p>В свободное время я люблю [ваше хобби].</p>
</body>
</html>
```

Сохраните файл и откройте в браузере

Проанализируйте результат:

Посмотрите на вкладку браузера — видите "имя" Вашей страницы?
Посмотрите на саму страницу — видите "речь" и "внешний вид"?

Заключение: Почему эта аналогия так полезна?

Эта простая аналогия "*Человек = Веб-страница*" помогает:

- Запомнить назначение тегов:** Голова для мыслей, тело для действий
- Понять структуру документа:** Что куда помещается
- Избежать ошибок:** Не пытаться поместить <title> в тело или картинку в голову
- Легко объяснять другим:** Теперь Вы можете понятно рассказать о структуре HTML любому человеку

Запомните эту аналогию, и Вы никогда не перепутаете, что должно быть в <head>, а что — в <body>. Ваша страница теперь не просто код, а целостная личность со своим именем и содержанием!

3.4. Практика: пишем "скелет" нашей первой страницы.

Введение: Собираем фундамент будущего сайта

Пришло время применить все полученные знания на практике! В этом уроке мы соберём идеальный "скелет" HTML-страницы. Это будет наш шаблон, который мы сможем использовать для всех будущих страниц. Помните аналогию с человеком? Сейчас мы создадим его кости, без которых невозможно двигаться дальше.

Часть 1: Подготовка рабочего пространства

Шаг 1: Откройте папку проекта

Найдите на Рабочем столе папку "**Мой первый сайт**"
Откройте её двойным щелчком мыши

Шаг 2: Создайте новый файл для практики

Щёлкните правой кнопкой мыши в папке
Выберите "**Создать**" → "**Текстовый документ**"

Сразу дайте файлу имя: skeleton.html
Нажмите Enter и подтвердите изменение расширения

Шаг 3: Откройте файл в Блокноте

Щёлкните по skeleton.html правой кнопкой мыши
Выберите "**Открыть с помощью**" → "**Блокнот**"
Теперь Вы видите чистый лист — наше "цифровое полотно" для создания скелета.

Часть 2: Пошаговое создание скелета

Давайте будем собирать наш скелет как конструктор, последовательно добавляя каждый элемент.

Шаг 1: Добавляем декларацию DOCTYPE

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

Пояснение: Эта строка сообщает браузеру, что мы используем современный стандарт HTML5

Действие: Введите эту строку в самой первой строке Блокнота

Шаг 2: Открываем основной контейнер

html

<html>

Пояснение: Это начало нашего HTML-документа, "тело" нашего виртуального человека

Действие: Введите эту строку второй

Шаг 3: Создаём раздел "головы"

html

<head>

</head>

Пояснение: В этом разделе будет служебная информация

Действие: Введите тег <head>, сделайте несколько пустых строк и закройте тег </head>

Совет: Сразу пишите парные теги, чтобы не забыть закрывающий!

Шаг 4: Добавляем мета-информацию в "голову"

Между тегами <head> и </head> добавьте:

html

<head>

<title>Мой первый скелет страницы</title>

<meta charset="utf-8">

</head>

Шаг 5: Создаём раздел "тела"

html

<body>

</body>

Пояснение: Здесь будет видимое содержимое страницы

Действие: Добавьте теги `<body>` и `</body>` после закрывающего тега `</head>`

Шаг 6: Закрываем основной контейнер

```
html  
</html>
```

Действие: Добавьте эту строку последней

Часть 3: Полный код скелета

Вот что у Вас должно получиться в Блокноте:

```
html  
<!DOCTYPE html>  
<html>  
<head>  
<title>Мой первый скелет страницы</title>  
<meta charset="utf-8">  
</head>  
<body>  
  
</body>  
</html>
```

Визуализируем структуру:

text
ДОКУМЕНТ
└─ <!DOCTYPE html> (ОБЪЯВЛЕНИЕ)
 └─ <html> (ВЕСЬ ДОКУМЕНТ)
 └─ <head> (СЛУЖЕБНАЯ ЧАСТЬ)
 └─ <title>...</title> (ЗАГОЛОВОК ВКЛАДКИ)
 └─ <meta charset="utf-8"> (КОДИРОВКА)
 └─ <body> (ВИДИМАЯ ЧАСТЬ)
 └─ [ЗДЕСЬ БУДЕТ КОНТЕНТ]

Часть 4: Проверяем и тестируем работу

Шаг 1: Сохраните файл

Нажмите **Ctrl + S** или выберите "Файл" → "Сохранить"

Шаг 2: Откройте в браузере

Вернитесь в папку "Мой первый сайт"
Дважды щёлкните по файлу skeleton.html

Шаг 3: Проанализируйте результат

Вы должны увидеть:

Во вкладке браузера: "Мой первый скелет страницы"

На самой странице: Абсолютно пустое белое поле

Поздравляю! Вы только что создали идеально структурированную HTML-страницу. То, что она пустая — нормально! Мы создали каркас, теперь можно наполнять его содержимым.

Часть 5: Добавляем первый контент

Давайте "оживим" нашего скелета, добавив простой контент.

Вернитесь в Блокнот и добавьте в раздел `<body>`:

```
html
<body>
<h1>Добро пожаловать на мою страницу!</h1>
<p>Это мой первый правильно созданный HTML-документ.</p>
<p>Я научился создавать каркас страницы!</p>
</body>
```

Сохраните файл (Ctrl + S) и обновите страницу в браузере (F5).

Теперь Вы увидите:

Крупный заголовок "Добро пожаловать на мою страницу!"

Два абзаца текста под ним

Часть 6: Практическое задание для закрепления

Задание: Создайте собственную страницу-визитку

Создайте новый файл vizitka.html в папке "Мой первый сайт"

Используя созданный скелет, заполните его следующим содержанием:

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

```
<title>Визитная карточка [Ваше Имя]</title>
<meta charset="utf-8">
</head>
<body>
<h1>[Ваше Имя и Фамилия]</h1>
<p>Мои увлечения: [перечислите 2-3 увлечения]</p>
<p>Мой опыт в IT: [опишите Ваш опыт]</p>
<p>Я учу HTML потому что: [Ваша мотивация]</p>
</body>
</html>
```

Сохраните и откройте в браузере

Убедитесь, что:
Заголовок вкладки отображается правильно.
Все абзацы текста показываются на странице.
Русские буквы отображаются без искажений.

Часть 7: Возможные проблемы и решения

Проблема 1: Браузер показывает код вместо текста

Причина: Вы забыли сохранить файл после внесения изменений.

Решение: Нажмите Ctrl + S в Блокноте и обновите страницу в браузере.

Проблема 2: Русские буквы отображаются как "кракозябры" (не читаемые символы)

Причина: Отсутствует или неправильно указана кодировка.

Решение: Убедитесь, что в разделе `<head>` есть строка `<meta charset="utf-8">`.

Проблема 3: Страница выглядит не как заголовок и абзацы, а как сплошной текст

Причина: Опечатка в тегах или забыли закрывающие теги.

Решение: Внимательно проверьте — каждый тег должен иметь парный закрывающий тег.

Заключение

Вы успешно создали свой первый HTML-скелет! Теперь Вы умеете:

- Создавать правильно структурированный HTML-документ
- Понимать назначение каждого обязательного тега
- Добавлять базовый контент на страницу
- Проверять результат в браузере

Этот скелет — Ваш главный шаблон. Все будущие страницы будут начинаться именно с этой структуры. В следующих уроках мы будем "наращивать мускулы" на этом скелете, добавляя списки, картинки, ссылки и таблицы.

Сохраните этот код — он станет Вашим надежным фундаментом для всех будущих веб-страниц!

Глава 4: Добавляем текст на страницу

4.1. Теги для заголовков: <h1>, <h2>, <h3> (как заголовки в газете)

Введение: Создаём иерархию и структуру

Представьте, что Вы открыли газету. Первое, что бросается в глаза — это крупные, заметные заголовки. Они сразу дают понять, о чём статья, какие в ней есть основные разделы и подразделы. В HTML для этой цели существуют теги заголовков — от <h1> до <h6>. Они помогают создать чёткую иерархию содержимого, делая Вашу страницу удобной для чтения как людям, так и поисковым системам.

Часть 1: Что такое теги заголовков и зачем они нужны?

Теги заголовков — это парные HTML-теги, которые используются для создания заголовков разного уровня важности.

Их основные задачи:

Структурирование контента: Разбивают текст на логические разделы.

Визуальное выделение: Браузер автоматически отображает их более крупным и жирным шрифтом.

Улучшение доступности: Специальные программы для слабовидящих используют заголовки для навигации по странице.

SEO (поисковая оптимизация): Поисковые системы (Яндекс, Google) обращают особое внимание на заголовки, чтобы понять, о чём Ваша страница.

Часть 2: Иерархия заголовков — от самого важного к менее важному

Существует 6 уровней заголовков, но на практике чаще всего используются <h1>, <h2> и <h3>.

Аналогия с газетной статьей

Уровень	Тег	Роль в газете	Роль на веб-странице
1 уровень	<h1>	Название всей газеты или главный заголовок на первой полосе.	Главный заголовок всей страницы. Должен быть один на странице.

2 уровень	<h2>	Названия крупных статей или разделов внутри статьи.	Заголовки основных разделов вашей страницы.
3 уровень	<h3>	Подзаголовки внутри статьи	Заголовки подразделов внутри основных разделов.
4-6 уровни	<h4>, <h5>, <h6>	Мелкие подпункты, сноски	Используются редко, для очень глубокой вложенности.

Часть 3: Визуальное представление заголовков

Давайте посмотрим, как браузер по умолчанию отображает разные уровни заголовков:

html

```
<h1>Это заголовок 1 уровня - самый крупный и важный</h1>
<h2>Это заголовок 2 уровня - поменьше</h2>
<h3>Это заголовок 3 уровня - ещё меньше</h3>
<h4>Это заголовок 4 уровня</h4>
<h5>Это заголовок 5 уровня</h5>
<h6>Это заголовок 6 уровня - самый мелкий</h6>
```

В браузере это будет выглядеть примерно так:

Это заголовок 1 уровня - самый крупный и важный

Это заголовок 2 уровня - поменьше

Это заголовок 3 уровня - ещё меньше

Это заголовок 4 уровня

Это заголовок 5 уровня

Это заголовок 6 уровня - самый мелкий

Часть 4: Практический пример — структура страницы "Моё хобби"

Давайте создадим правильно структурированную страницу, используя заголовки разных уровней.

Неправильный подход (без иерархии):

html

```
<h1>Моё хобби - Садоводство</h1>
```

```
<h3>Выращивание овощей</h3>
<h2>Помидоры</h2>
<h3>Цветы в моём саду</h3>
<h2>Розы</h2>
```

Проблема: Иерархия нарушена. **<h3>** идет после **<h1>**, **<h2>** находится на одном уровне с **<h3>**.

Правильный подход (логическая иерархия):

html

```
<h1>Моё хобби - Садоводство</h1> <!-- Главная тема страницы -->

<h2>Выращивание овощей</h2> <!-- Крупный раздел -->
<h3>Помидоры</h3> <!-- Подраздел внутри "Овощей" -->
<h3>Огурцы</h3> <!-- Еще один подраздел -->

<h2>Цветы в моём саду</h2> <!-- Другой крупный раздел -->
<h3>Розы</h3> <!-- Подраздел внутри "Цветов" -->
<h3>Тюльпаны</h3> <!-- Еще один подраздел -->
```

Часть 5: Практическое задание

Задание: Создайте страницу "Любимые рецепты" с правильной иерархией

Создайте новый файл `recepty.html` в Вашей папке "Мой первый сайт".

Используйте базовый скелет из предыдущего урока.

Наполните раздел **<body>** следующим содержанием:

```
html
<!DOCTYPE html>
<html>
<head>
<title>Мои любимые рецепты</title>
<meta charset="utf-8">
</head>
<body>
<h1>Кулинарная книга начинающего повара</h1>

<h2>Выпечка</h2>
<h3>Блины классические</h3>
<h3>Шарлотка с яблоками</h3>
```

```
<h2>Салаты</h2>
<h3>Оливье по-семейному</h3>
<h3>Винегрет овощной</h3>

<h2>Горячие блюда</h2>
<h3>Жаркое в горшочках</h3>
<h3>Курица с картошкой в духовке</h3>
</body>
</html>
```

Сохраните файл и откройте его в браузере.

Проанализируйте результат:

Убедитесь, что **<h1>** самый крупный
Проверьте, что **<h2>** меньше, чем **<h1>**, но больше, чем **<h3>**
Обратите внимание на автоматические отступы между заголовками

Часть 6: Важные правила и частые ошибки

ПРАВИЛЬНО:

Использовать **только один <h1>** на странице
Соблюдать иерархию: после **<h1>** должен идти **<h2>**, а не **<h3>**
Использовать заголовки по смыслу, а не для красоты

НЕПРАВИЛЬНО:

Использовать несколько **<h1>** на одной странице
Пропускать уровни (например, после **<h1>** сразу **<h3>**)
Использовать заголовки вместо обычного текста только чтобы сделать его крупнее

Ошибка новичков:

```
html
<!-- НЕПРАВИЛЬНО! -->
<h1>Главная страница</h1>
<h1>Новости</h1> <!-- Два h1 на одной странице! -->
<h3>Сегодня хорошая погода</h3> <!-- Пропущен уровень h2! -->
```

Часть 7: Дополнительные возможности

Заголовки с выравниванием (устаревший способ, но полезно знать):

```
html
<h1 align="center">Заголовок по центру</h1>
```

<h2 align="right">Заголовок справа</h2>

Примечание: Современный способ выравнивания — с помощью CSS⁴, но для начала можно использовать и этот.

Заключение

Теги заголовков <h1>, <h2>, <h3> — это Ваш главный инструмент для создания логичной и понятной структуры веб-страницы.

Запомните:

<h1> — главный заголовок (один на страницу)

<h2> — разделы

<h3> — подразделы

Теперь Ваши страницы будут не просто набором текста, а хорошо структурированными документами, похожими на газетные статьи с чёткой иерархией. Это сделает их более удобными для чтения и более "понятными" для поисковых систем.

Поздравляю! Вы освоили один из самых важных инструментов HTML-верстки!

⁴ CSS – каскадные таблицы стилей

4.2. Абзац текста: тег `<p>`

Введение: Основы читабельного текста

Представьте книгу, в которой весь текст идет одной сплошной прописью без абзацев. Читать такое практически невозможно! Тег `<p>` — это решение этой проблемы в HTML. Он создаёт логические и визуальные абзацы, делая Ваш текст удобным для восприятия.

Часть 1: Что такое тег `<p>` и зачем он нужен?

`<p>` (от английского "paragraph" — абзац) — это парный HTML-тег, который используется для оформления обычного текста в виде абзацев.

Основные задачи тега `<p>`:

Логическое разделение текста на смысловые блоки

Визуальное оформление — браузер автоматически добавляет отступы

Улучшение читаемости текста для пользователей

Семантическая разметка — помочь поисковым системам в понимании структуры контента

Часть 2: Как работает тег `<p>`?

Базовый синтаксис:

html

```
<p>Здесь находится текст Вашего абзаца.</p>
```

Визуальный эффект:

Браузер автоматически добавляет:

Верхний и нижний отступ (как бы "воздух" вокруг абзаца)

Выравнивание по левому краю

Перенос строк по ширине окна браузера

Часть 3: Сравнение — с абзацами и без

✗ БЕЗ тегов `<p>` (неправильно):

html

<body>

Сегодня я хочу рассказать о своём увлечении.

Я очень люблю садоводство.

Это началось ещё в детстве, когда бабушка учила меня ухаживать за растениями.

Сейчас у меня есть небольшой сад, где я выращиваю цветы и овощи.

</body>

Результат:

Сегодня я хочу рассказать о своём увлечении. Я очень люблю садоводство. Это началось ещё в детстве, когда бабушка учila меня ухаживать за растениями. Сейчас у меня есть небольшой сад, где я выращиваю цветы и овощи.

Весь текст сливаются в одну строку, читать очень неудобно.

С тегами <p> (правильно):

html

<body>

<p>Сегодня я хочу рассказать о своём увлечении.</p>

<p>Я очень люблю садоводство.</p>

<p>Это началось ещё в детстве, когда бабушка учila меня ухаживать за растениями.</p>

<p>Сейчас у меня есть небольшой сад, где я выращиваю цветы и овощи.</p>

</body>

Результат: Каждое предложение становится отдельным абзацем с отступами.

Сегодня я хочу рассказать о своём увлечении.

Я очень люблю садоводство.

Это началось ещё в детстве, когда бабушка учila меня ухаживать за растениями.

Сейчас у меня есть небольшой сад, где я выращиваю цветы и овощи.

Часть 4: Практические примеры использования

Пример 1: Простые абзацы

html

```
<p>Добро пожаловать на мою страницу!</p>
<p>Меня зовут Анна, и я создаю свой первый сайт.</p>
<p>Надеюсь, у меня всё получится.</p>
```

Пример 2: Абзацы в сочетании с заголовками

html

```
<h1>Моё хобби - Вязание</h1>
<p>Вязанием я увлеклась ещё в молодости. Бабушка научила меня
основам, и с тех пор это стало моим любимым занятием.</p>

<h2>Что я люблю вязать</h2>
<p>Чаще всего я вяжу тёплые свитера и носки для внуков. Они всегда
радуются новым вещам.</p>
<p>Также я освоила вязание игрушек - получаются забавные мишки и
зайчики.</p>
```

Пример 3: Длинный текст с абзацами

html

```
<h1>История моего сада</h1>
<p>Всё началось пять лет назад, когда мы купили дачный участок.
Земля была совершенно пустой - только трава и несколько старых
деревьев.</p>
<p>Первым делом мы разбили грядки для овощей. Посадили помидоры,
огурцы, морковь и зелень. Первый урожай был небольшим, но очень
радостным.</p>
<p>На следующий год я решила заняться цветами. Посадила розы,
пионы и тюльпаны. Теперь каждое лето мой сад утопает в цветах!</p>
```

Часть 5: Практическое задание

Задание: Создайте страницу "О себе" с использованием абзацев

Создайте новый файл about.html в папке "Мой первый сайт"

Используйте базовый скелет страницы

Наполните раздел <body> содержанием:

```
html
<!DOCTYPE html>
<html>
<head>
<title>Обо мне</title>
<meta charset="utf-8">
</head>
<body>
<h1>Немного о себе</h1>

<p>Здравствуйте! Меня зовут Виктор, и я пенсионер.</p>

<p>Всю жизнь я проработал инженером на заводе. Сейчас на пенсии решил освоить что-то новое и выбрал создание сайтов.</p>

<p>Мне всегда нравилось разбираться в технике, а компьютеры особенно интересны. Дети подарили мне ноутбук, и теперь я учусь работать в интернете.</p>

<p>В свободное время люблю читать книги, работать в саду и общаться с внуками. Они помогают мне осваивать новые технологии.</p>
</body>
</html>
```

Сохраните файл и откройте в браузере

Проанализируйте результат:

Убедитесь, что между абзацами есть отступы
Проверьте, что текст автоматически переносится по ширине окна
Попробуйте изменить размер окна браузера - текст будет перестраиваться

Часть 6: Важные особенности и правила

КАК ПРАВИЛЬНО ИСПОЛЬЗОВАТЬ:

Всегда закрывайте тег — <p> является парным тегом

Используйте для логических абзацев — не для отдельных строк

Сочетайте с заголовками для лучшей структуры

Не злоупотребляйте — слишком короткие абзацы выглядят неестественно

ЧАСТЫЕ ОШИБКИ:

html

```
<!-- НЕПРАВИЛЬНО: не закрыт тег -->
<p>Какой-то текст
```

```
<!-- НЕПРАВИЛЬНО: тег p для одной строки -->
<p>Имя: Иван</p>
<p>Фамилия: Петров</p>
<p>Возраст: 65 лет</p>
```

```
<!-- НЕПРАВИЛЬНО: вложенные абзацы -->
<p>
Первый абзац
<p>Вложенный абзац</p>
</p>
```

Часть 7: Полезные советы для пенсионеров

Не бойтесь делать ошибки — всегда можно исправить

Сначала пишите текст в Блокноте без тегов, потом размечайте абзацы

Используйте отступы в коде для удобства чтения:

html

```
<body>
<h1>Заголовок</h1>

<p>Первый абзац</p>

<p>Второй абзац</p>
</body>
```

Проверяйте результат после каждого добавленного абзаца

Часть 8: Дополнительные возможности

Выравнивание текста в абзаце:

html

```
<p align="left">Текст слева (по умолчанию)</p>
<p align="center">Текст по центру</p>
<p align="right">Текст справа</p>
<p align="justify">Текст по ширине</p>
```

Абзац с переносами строк внутри:

html

```
<p>Первая строка абзаца.<br>
Вторая строка того же абзаца.<br>
Третья строка.</p>
```

Заключение

Тег `<p>` — это один из самых часто используемых и важных тегов в HTML. Он превращает бесформенный текст в удобочитаемые абзацы.

Что Вы теперь умеете:

- Создавать логические абзацы текста
- Правильно структурировать содержимое страницы
- Делать текст удобным для чтения
- Сочетать заголовки и абзацы

Запомните: Хорошо структурированный текст — это залог того, что Вашу страницу будут читать с удовольствием. Тег `<p>` — Ваш главный помощник в этом деле!

Теперь Ваши веб-страницы станут по-настоящему читабельными и профессионально выглядящими. Поздравляю с освоением этого важного элемента HTML!

4.3. Перенос строки
: когда нужно просто перейти на новую строку

Введение: Контролируем расположение текста

Иногда нам нужно просто перейти на новую строку, не начиная новый абзац. Представьте, что Вы пишете адрес или стихотворение — каждая часть должна быть с новой строки, но при этом это единое целое. Для таких случаев в HTML существует специальный тег
.

Часть 1: Что такое тег
 и зачем он нужен?

 (от английского "break" — разрыв) — это одиночный HTML-тег, который принудительно переносит текст на следующую строку.

Ключевые особенности:

Это **одиночный тег** — у него нет закрывающего тега </br>

Создает **одиночный перенос строки**

Не добавляет вертикальных отступов (в отличие от <p>)

Часть 2: Отличие
 от тега <p>

Давайте наглядно сравним эти два тега:

Тег <p> (для абзацев):

```
html
<p>Это первый абзац.</p>
<p>Это второй абзац.</p>
```

Результат:

Это первый абзац.

Это второй абзац.

Между абзацами есть заметные отступы

Тег
 (для переносов внутри абзаца):

```
html
<p>Это первая строка.<br>Это вторая строка.<br>Это третья строка.</p>
```

Результат:

```
|  
|  
|   Это первая строка.  
|   Это вторая строка.  
|   Это третья строка.  
|
```

Строки идут друг за другом без отступов между ними

Часть 3: Когда использовать
 — практические примеры

Пример 1: Адрес

html

```
<p>Мой адрес:<br>  
ул. Садовая, дом 15<br>  
город Москва<br>  
123456</p>
```

Результат:

```
|  
|  
|   Мой адрес:  
|   ул. Садовая, дом 15  
|   город Москва  
|   123456  
|
```

Пример 2: Стихотворение

html

```
<p>Уронили мишку на пол,<br>  
Оторвали мишке лапу.<br>  
Всё равно его не брошу —<br>  
Потому что он хороший.</p>
```

Пример 3: Контактная информация

html

```
<p>Иванов Пётр Сергеевич<br>  
Телефон: +7 (123) 456-78-90<br>  
Email: petr@example.com</p>
```

Пример 4: Подпись под фотографией

html

```
<p>Наша дача в прошлое воскресенье.<br>
Фото сделано 15 мая 2024 года.</p>
```

Часть 4: Частые ошибки и как их избежать

НЕПРАВИЛЬНО:

Ошибка 1: Использование
 вместо абзацев

html

```
Текст первой мысли.<br><br>
Текст второй мысли.<br><br>
Текст третьей мысли.
```

Проблема: Два тега
 подряд создают имитацию абзаца, но это неправильно с семантической точки зрения.

Ошибка 2: Закрывающий тег

html

```
Первая строка<br></br>Вторая строка
```

Проблема: Тег
 не требует закрывающего тега.

ПРАВИЛЬНО:

Для абзацев используйте <p>:

html

```
<p>Текст первой мысли.</p>
<p>Текст второй мысли.</p>
<p>Текст третьей мысли.</p>
```

Для переносов внутри абзаца используйте
:

html

```
<p>Первая строка<br>Вторая строка<br>Третья строка</p>
```

Часть 5: Практическое задание

Задание: Создайте страницу с контактной информацией

Создайте новый файл contacts.html в папке "Мой первый сайт"

Напишите код:

```
html
<!DOCTYPE html>
<html>
<head>
<title>Мои контакты</title>
<meta charset="utf-8">
</head>
<body>
<h1>Как со мной связаться</h1>
```

```
<h2>Почтовый адрес</h2>
<p>Иванова Мария Петровна<br>
ул. Цветочная, дом 25, кв. 12<br>
город Санкт-Петербург<br>
190000</p>
```

```
<h2>Электронные контакты</h2>
<p>Телефон: +7 (911) 123-45-67<br>
Email: maria.ivanova@example.com<br>
Скайп: maria.ivanova</p>
```

```
<h2>График работы</h2>
<p>Понедельник-пятница: 9:00-18:00<br>
Суббота: 10:00-16:00<br>
Воскресенье: выходной</p>
```

```
<h2>Любимое стихотворение</h2>
<p>У меня есть дед любимый,<br>
Он такой у нас единственный!<br>
Знает тысяч сто историй,<br>
Добрый, умный и герой!</p>
</body>
</html>
```

Сохраните файл и откройте в браузере

Проанализируйте результат:

Убедитесь, что в адресе каждая часть на новой строке
Проверьте, что контактная информация читается легко
Обратите внимание, что стихотворение имеет правильное оформление

Часть 6: Полезные советы для начинающих

Запоминаем синтаксис:
 — самый короткий тег, пишется без закрывающего тега

Используем осмысленно: Только когда действительно нужен перенос строки внутри одного логического блока

Проверяем результат: После добавления
 обязательно смотрим в браузере

Не злоупотребляем: Множественные
 для создания больших отступов — плохая практика

Пример разумного использования:

```
html
<!-- ХОРОШО: 1-2 переноса для структуры -->
<p>Директору ООО "Ромашка"<br>
от Ивановой М.П.</p>
```

Пример избыточного использования:

```
html
<!-- ПЛОХО: много переносов для создания отступа -->
<p>Текст перед отступом<br><br><br><br><br>
Текст после отступа</p>
```

Часть 7: Дополнительная информация

XHTML синтаксис:

В более строгом XHTML тег
 записывается как самозакрывающийся:

```
html
<br />
```

В обычном HTML5 оба варианта работают, но
 предпочтительнее.

Невидимый тег:

Тег
 не имеет видимого представления сам по себе — он только переносит текст на следующую строку.

Часть 8: Закрепляющее упражнение

Найдите и исправьте ошибки в коде:

```
html
<!DOCTYPE html>
<html>
<head>
<title>Упражнение</title>
</head>
<body>
<p>Первая строка<br>
Вторая строка</br>
Третья строка<br><br>
Четвертая строка</p>

<p>Это должен быть отдельный абзац<br>
Но здесь лишний перенос</p>
</body>
</html>
```

Правильный вариант:

```
html
<!DOCTYPE html>
<html>
<head>
<title>Упражнение</title>
</head>
<body>
<p>Первая строка<br>
Вторая строка<br>
Третья строка<br>
Четвертая строка</p>

<p>Это должен быть отдельный абзац</p>
<p>Но здесь лишний перенос</p>
</body>
</html>
```

Заключение

Тег `
` — это простой, но очень полезный инструмент для точного контроля над расположением текста.

Что Вы теперь умеете:

- Создавать переносы строк внутри абзацев
- Правильно оформлять адреса и стихи

- Отличать ситуации для
 и <p>
- Избегать частых ошибок новичков

Золотое правило:

Используйте <p> для логических абзацев (с отступами)

Используйте
 для принудительных переносов внутри абзаца (без отступов)

Теперь Вы можете создавать тексты любой сложности с точным контролем над их внешним видом! Поздравляю с освоением этого полезного тега!

4.4. Горизонтальная линия <hr> для разделения содержимого

Введение: Визуальный разделитель контента

Представьте книгу, где главы разделены не только заголовками, но и декоративными элементами. Тег <hr> (от "horizontal rule" — горизонтальная линия) выполняет именно эту роль в HTML — он создает визуальный разделитель между различными блоками контента.

Часть 1: Что такое тег <hr> и зачем он нужен?

<hr> — это одиночный HTML-тег, который создает горизонтальную линию-разделитель.

Основные задачи:

Визуальное разделение смысловых блоков контента

Акцентирование внимания на смене темы

Улучшение читаемости за счет структурирования

Декоративное оформление страницы

Особенности:

Это **одиночный тег** (не требует закрывающего </hr>)

По умолчанию занимает **всю доступную ширину**

Имеет **небольшие отступы** сверху и снизу

Отображается как **простая серая линия**

Часть 2: Базовое использование тега <hr>

Простой пример:

```
html
<p>Текст первого раздела</p>
<hr>
<p>Текст второго раздела</p>
```

Результат:

Текст первого раздела

Текст второго раздела

Пример с заголовками:

```
html
<h2>Мои достижения</h2>
<p>В этом году я вырастил лучший урожай помидоров...</p>

<hr>

<h2>Планы на будущее</h2>
<p>В следующем сезоне хочу попробовать вырастить арбузы...</p>
```

Мои достижения

В этом году я вырастил лучший урожай помидоров...

Планы на будущее

Часть 3: В следующем сезоне хочу попробовать вырастить арбузы...

**Внешний
вид <hr>**

> по умолчанию в разных браузерах

Браузеры могут немного по-разному отображать линию:

Браузер	Толщина	Цвет	Стиль	Отступы
Chrome	1px	#ccc	inset	margin: 0.5em 0
Firefox	1px	#ccc	inset	margin: 0.5em 0
Edge	1px	#ccc	inset	margin: 0.5em 0

Часть 4: Практические примеры использования

Пример 1: Разделение разделов статьи

```
html
<article>
<h1>Выращивание роз для начинающих</h1>
<p>Розы — одни из самых красивых садовых цветов...</p>
```

```
<hr>

<h2>Выбор места для посадки</h2>
<p>Розы любят солнечные, защищенные от ветра места...</p>

<hr>

<h2>Полив и уход</h2>
<p>Поливайте розы утром или вечером, избегая попадания воды на листья...</p>
</article>
```

Выращивание роз для начинающих

Розы — одни из самых красивых садовых цветов...

Выбор места для посадки

Розы любят солнечные, защищенные от ветра места...

Полив и уход

Поливайте розы утром или вечером, избегая попадания воды на листья...

Пример 2: Разделение комментариев

```
html
<div class="comment">
<p><strong>Мария:</strong> Отличная статья! Спасибо за советы.</p>
</div>

<hr>

<div class="comment">
<p><strong>Иван:</strong> Попробовал Ваши рекомендации —
работает!</p>
</div>

<hr>

<div class="comment">
```

```
<p><strong>Ольга:</strong> А когда лучше обрезать розы?</p>
</div>
```

Мария: Отличная статья! Спасибо за советы.

Иван: Попробовал Ваши рекомендации — работает!

Ольга: А когда лучше обрезать розы?

Пример 3: Разделение контактной информации

html

```
<h2>Контактная информация</h2>
<p>Телефон: +7 (123) 456-78-90</p>
<p>Email: example@mail.ru</p>
```

```
<hr>
```

```
<h2>График работы</h2>
<p>Пн-Пт: 9:00-18:00</p>
<p>Сб: 10:00-16:00</p>
<p>Вс: выходной</p>
```

Контактная информация

Телефон: +7 (123) 456-78-90

Email: example@mail.ru

График работы

Пн-Пт: 9:00-18:00

Сб: 10:00-16:00

Вс: выходной

Часть 5: Стилизация линии с помощью CSS

Вы можете легко изменить внешний вид линии:

Изменение толщины и цвета:

html

```
<hr style="border: 2px solid red;">
```

|

Контактная информация

Телефон: +7 (123) 456-78-90

Email: example@mail.ru

График работы

Пн-Пт: 9:00-18:00

Сб: 10:00-16:00

Вс: выходной

Пунктирная линия:

html

```
<hr style="border: 1px dashed blue;">
```

|

Контактная информация

Телефон: +7 (123) 456-78-90

Email: example@mail.ru

График работы

Пн-Пт: 9:00-18:00

Сб: 10:00-16:00

Вс: выходной

Тонкая серая линия:

```
html
<hr style="border: none; height: 1px; background-color: #eee;">
```

Контактная информация

Телефон: +7 (123) 456-78-90

Email: example@mail.ru

График работы

Пн-Пт: 9:00-18:00

Сб: 10:00-16:00

Вс: **выходной**

Цветная линия с градиентом:

```
html
<hr style="border: none; height: 3px; background: linear-gradient(to right, red,
orange, yellow);">
```

Контактная информация

Телефон: +7 (123) 456-78-90

Email: example@mail.ru

График работы

Пн-Пт: 9:00-18:00

Сб: 10:00-16:00

Вс: выходной

Линия с тенью:

html

```
<hr style="border: none; height: 2px; background: #333; box-shadow: 0 2px  
4px rgba(0,0,0,0.3);">
```

Контактная информация

Телефон: +7 (123) 456-78-90

Email: example@mail.ru

График работы

Пн-Пт: 9:00-18:00

Сб: 10:00-16:00

Вс: выходной

Часть 6: Практическое задание

Задание: Создайте страницу с различными типами разделителей

html

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

```
<head>
<title>Различные разделители</title>
<meta charset="utf-8">
<style>
.section {
margin: 20px 0;
padding: 15px;
border: 1px solid #ddd;
}
</style>
</head>
<body>
<h1>Демонстрация различных разделителей</h1>

<div class="section">
<h2>1. Стандартная линия</h2>
<p>Это обычная горизонтальная линия по умолчанию.</p>
<hr>
<p>Текст после стандартной линии.</p>
</div>

<div class="section">
<h2>2. Толстая цветная линия</h2>
<p>Линия с пользовательскими стилями.</p>
<hr style="border: 3px solid #4CAF50;">
<p>Текст после зеленой линии.</p>
</div>

<div class="section">
<h2>3. Пунктирная линия</h2>
<p>Стилизованная пунктирная линия.</p>
<hr style="border: none; border-top: 2px dashed #ff6b6b;">
<p>Текст после пунктирной линии.</p>
</div>

<div class="section">
<h2>4. Градиентная линия</h2>
<p>Декоративная линия с градиентом.</p>
<hr style="border: none; height: 4px; background: linear-gradient(to right, #667eea, #764ba2);">
<p>Текст после градиентной линии.</p>
</div>
</body>
</html>
```

Демонстрация различных разделителей

1. Стандартная линия

Это обычная горизонтальная линия по умолчанию.

Текст после стандартной линии.

2. Толстая цветная линия

Линия с пользовательскими стилями.



Текст после зеленой линии.

3. Пунктирная линия

Стилизованная пунктирная линия.



Текст после пунктирной линии.

4. Градиентная линия

Декоративная линия с градиентом.



Текст после градиентной линии.

Часть 7: Семантическое значение <hr>

В HTML5 тег `<hr>` получил семантическое значение — он обозначает **тематический разрыв** на уровне абзацев.

Правильное использование:

html

```
<!-- ПРАВИЛЬНО: тематический разрыв -->
<article>
<p>Обсуждаем выращивание овощей...</p>
<hr>
<p>Теперь поговорим о цветах...</p>
</article>
```

Неправильное использование:

```
html
<!-- НЕПРАВИЛЬНО: чисто декоративное использование -->
<div class="header">
<h1>Заголовок</h1>
<hr>
<p>Подзаголовок</p>
</div>
```

Часть 8: Альтернативы тегу <hr>

Иногда лучше использовать другие методы разделения:

CSS border:

```
html
<div style="border-bottom: 1px solid #ccc; padding-bottom: 10px; margin-
bottom: 10px;">
Первый блок
</div>
<div>
Второй блок
</div>
```

Пустой div с фоном:

```
html
<div style="height: 1px; background: #ccc; margin: 20px 0;"></div>
```

CSS псевдоэлементы:

```
html
<style>
.decorative-line::after {
content: "";
display: block;
```

```
height: 2px;  
background: linear-gradient(to right, transparent, #333, transparent);  
margin: 10px 0;  
}  
</style>
```

```
<div class="decorative-line">  
Текст с декоративной линией  
</div>
```

Часть 9: Частые ошибки и лучшие практики

ПРАВИЛЬНО:

```
html  
<!-- Разделение смысловых блоков --&gt;<br/><section>  
<h2>Введение</h2>  
<p>Текст введения...</p>  
</section>  
  
<hr>  
  
<section>  
<h2>Основная часть</h2>  
<p>Текст основной части...</p>  
</section>
```

НЕПРАВИЛЬНО:

```
html  
<!-- Слишком частые разделители --&gt;<br/><p>Первый абзац</p>  
<hr>  
<p>Второй абзац</p>  
<hr>  
<p>Третий абзац</p>  
<hr>  
  
<!-- Использование для декоративных целей --&gt;<br/><div class="product-card">  
<h3>Название товара</h3>  
<hr> <!-- Лучше использовать CSS border --&gt;<br/><p>Описание товара</p>  
</div>
```

Часть 10: Специальные эффекты с <hr>

Анимированная линия:

html

```
<hr style="border: none; height: 3px; background: #333;  
animation: pulse 2s infinite;">
```

```
<style>  
@keyframes pulse {  
0% { opacity: 0.3; }  
50% { opacity: 1; }  
100% { opacity: 0.3; }  
}  
</style>
```

Линия с иконкой:

html

```
<div style="text-align: center;">  
<hr style="display: inline-block; width: 40%; vertical-align: middle;">  
<span style="margin: 0 10px;">*</span>  
<hr style="display: inline-block; width: 40%; vertical-align: middle;">  
</div>
```

Заключение

Тег <hr> — это простой, но мощный инструмент для структурирования контента:

Что Вы теперь умеете:

- Создавать стандартные горизонтальные линии
- Стилизовать линии с помощью CSS
- Правильно использовать для семантического разделения
- Избегать частых ошибок новичков

Золотые правила использования:

Используйте для **тематического разделения** блоков контента

Не злоупотребляйте — слишком много линий мешает чтению

Для чисто **декоративных** целей лучше использовать CSS

Тестируйте внешний вид в разных браузерах

Теперь Ваши веб-страницы будут иметь четкую, хорошо структурированную визуальную иерархию!

4.5. Практика: создаём страницу-визитку о себе с заголовком и несколькими абзацами

Введение: Собираем всё вместе в практическом проекте

Пришло время применить все изученные теги на практике! Мы создадим настоящую веб-страницу-визитку, которую Вы сможете показать друзьям и родным. Это будет Ваш первый полноценный HTML-документ.

Часть 1: Подготовка к работе

Что нам понадобится:

Папка "Мой первый сайт" (уже создана ранее)

Блокнот или Notepad++

Браузер для просмотра результата

Создаем новый файл:

Откройте папку "Мой первый сайт"

Щелкните правой кнопкой мыши → "Создать" → "Текстовый документ"

Назовите файл: vizitka.html

Нажмите Enter и подтвердите изменение расширения

Часть 2: Создаём базовую структуру страницы

Откройте файл vizitka.html в Блокноте и напишите:

```
html
<!DOCTYPE html>
<html lang="ru">
<head>
<title>Моя персональная страница-визитка</title>
<meta charset="utf-8">
</head>
<body>
<!-- Здесь будет содержимое нашей визитки --&gt;
&lt;/body&gt;
&lt;/html&gt;</pre>
```

Пояснение структуры:

<!DOCTYPE html> — объявление типа документа

<html lang="ru"> — указываем язык страницы (русский)

<head> — служебная информация

```
<title> — заголовок вкладки браузера  
<meta charset="utf-8"> — поддержка русских букв  
<body> — видимое содержимое страницы
```

Часть 3: Наполняем страницу контентом

Добавьте в раздел `<body>` следующий код:

```
html  
<body>  
  <!-- Главный заголовок -->  
  <h1>Добро пожаловать на мою страницу!</h1>  
  
  <!-- Раздел "О себе" -->  
  <h2>Немного о себе</h2>  
  <p>Здравствуйте! Меня зовут Александр, и я начинающий веб-разработчик.</p>  
  <p>Недавно вышел на пенсию и решил освоить новую для себя профессию - создание сайтов. Мне всегда нравилось разбираться в технике, а компьютеры вызывали особый интерес.</p>  
  
  <!-- Раздел "Мои увлечения" -->  
  <h2>Мои увлечения</h2>  
  <p>В свободное время я люблю работать в саду, читать книги и проводить время с внуками.</p>  
  <p>Особенно мне нравится выращивать розы - у меня уже собралась небольшая коллекция из 15 разных сортов.</p>  
  
  <!-- Раздел "Почему я учу HTML" -->  
  <h2>Почему я учу HTML</h2>  
  <p>Я верю, что никогда не поздно учиться чему-то новому. Освоение веб-разработки помогает мне поддерживать умственную активность и идти в ногу со временем.</p>  
  <p>Моя цель - создать собственный сайт о садоводстве, где я смогу делиться своим опытом с другими любителями растений.</p>  
  
  <!-- Разделительная линия -->  
  <hr>  
  
  <!-- Контактная информация -->  
  <h3>Контактная информация</h3>  
  <p>Email: alexander@example.com</p>  
  <p>Телефон: +7 (123) 456-78-90</p>  
</body>
```

Часть 4: Полный код страницы-визитки

Вот как должен выглядеть полный код в Вашем Блокноте:

```
html
<!DOCTYPE html>
<html lang="ru">
<head>
<title>Моя персональная страница-визитка</title>
<meta charset="utf-8">
</head>
<body>
<h1>Добро пожаловать на мою страницу!</h1>

<h2>Немного о себе</h2>
<p>Здравствуйте! Меня зовут Александр, и я начинающий веб-разработчик.</p>
<p>Недавно вышел на пенсию и решил освоить новую для себя профессию - создание сайтов. Мне всегда нравилось разбираться в технике, а компьютеры вызывали особый интерес.</p>

<h2>Мои увлечения</h2>
<p>В свободное время я люблю работать в саду, читать книги и проводить время с внуками.</p>
<p>Особенно мне нравится выращивать розы - у меня уже собралась небольшая коллекция из 15 разных сортов.</p>

<h2>Почему я учу HTML</h2>
<p>Я верю, что никогда не поздно учиться чему-то новому. Освоение веб-разработки помогает мне поддерживать умственную активность и идти в ногу со временем.</p>
<p>Моя цель - создать собственный сайт о садоводстве, где я смогу делиться своим опытом с другими любителями растений.</p>

<hr>

<h3>Контактная информация</h3>
<p>Email: alexander@example.com</p>
<p>Телефон: +7 (123) 456-78-90</p>
</body>
</html>
```

Часть 5: Сохранение и просмотр результата

Сохраните файл в Блокноте: Ctrl + S

Откройте файл в браузере:

Вариант 1: Дважды щелкните по vizitka.html в папке

Вариант 2: Откройте браузер и нажмите Ctrl + O, затем выберите файл

Что Вы должны увидеть в браузере:

Крупный заголовок "Добро пожаловать на мою страницу!"

Несколько разделов с подзаголовками

Абзацы текста с отступами

Горизонтальную разделительную линию

Контактную информацию внизу

Часть 6: Анализируем структуру страницы

Давайте разберем, какие теги мы использовали и зачем:

Тег	Количество	Назначение
<h1>	1	Главный заголовок страницы
<h2>	3	Заголовки основных разделов
<h3>	1	Заголовок подраздела
<p>	7	Абзацы текста
<hr>	1	Разделительная линия

Визуальная иерархия:

text

h1: Добро пожаловать на мою страницу!

h2: Немного о себе

p: Здравствуйте! Меня зовут...

p: Недавно вышел на пенсию...

h2: Мои увлечения

p: В свободное время...

p: Особенно мне нравится...

h2: Почему я учу HTML

p: Я верю, что никогда...

p: Моя цель - создать...

hr: _____

h3: Контактная информация

p: Email: ...

p: Телефон: ...

Часть 7: Персонализируйте свою визитку

Замените примерный текст на СВОЙ собственный:

```
html
<!DOCTYPE html>
<html lang="ru">
<head>
<title>Персональная страница [Ваше Имя]</title>
<meta charset="utf-8">
</head>
<body>
<h1>Добро пожаловать на мою страницу!</h1>

<h2>Обо мне</h2>
<p>Меня зовут [Ваше Имя]. [Несколько предложений о себе].</p>
<p>[Расскажите о своем прошлом опыте и почему решили учить
HTML].</p>

<h2>Мои интересы</h2>
<p>[Опишите Ваши основные увлечения и хобби].</p>
<p>[Расскажите о самом интересном увлечении подробнее].</p>

<h2>Мои цели в изучении HTML</h2>
<p>[Почему Вы решили учить веб-разработку?].</p>
<p>[Какую пользу Вы надеетесь извлечь из этих знаний?].</p>

<hr>

<h3>Как со мной связаться</h3>
<p>Email: [ваш email]</p>
<p>Телефон: [ваш телефон]</p>
</body>
</html>
```

Часть 8: Улучшаем визитку — дополнительные элементы

Добавьте эти элементы для большей наглядности:

Используем **
** для адреса:

```
html
<h3>Мой адрес</h3>
<p>ул. Садовая, дом 15<br>
```

город Москва

123456</p>

Добавляем дополнительные разделители:

html
<h2>Мои увлечения</h2>
<p>Садоводство и цветоводство</p>
<hr style="width: 50%; margin: 10px auto;">
<p>Чтение исторических книг</p>
<hr style="width: 50%; margin: 10px auto;">
<p>Фотография природы</p>

Используем специальные символы:

html
<p>Любимое время года: весна • Любимый цветок: роза ♥</p>

Часть 9: Проверка и отладка

После сохранения файла проверьте:

Открывается ли страница в браузере?

Правильно ли отображаются русские буквы?

Если видны "кракозябры", проверьте наличие <meta charset="utf-8">

Соответствует ли структура Вашим ожиданиям?

Заголовки должны быть крупнее обычного текста
Между абзацами должны быть отступы
Линия должна разделять блоки

Частые ошибки и их решение:

Текст сливаются в одну строку — забыли теги <p>

Не отображается заголовок вкладки — проверьте тег <title>

Неправильная иерархия заголовков —
используйте <h1>, <h2>, <h3> по порядку

Часть 10: Дополнительное задание для практики

Создайте вторую страницу "Мои рецепты":

```
html
<!DOCTYPE html>
<html lang="ru">
<head>
<title>Мои любимые рецепты</title>
<meta charset="utf-8">
</head>
<body>
<h1>Кулинарная книга</h1>

<h2>Выпечка</h2>
<h3>Блины классические</h3>
<p>Мука - 2 стакана<br>
Молоко - 500 мл<br>
Яйца - 2 штуки<br>
Сахар - 2 столовые ложки</p>

<h3>Шарлотка с яблоками</h3>
<p>Яблоки - 5 штук<br>
Мука - 1 стакан<br>
Яйца - 3 штуки<br>
Сахар - 1 стакан</p>

<hr>

<h2>Салаты</h2>
<h3>Оливье по-семейному</h3>
<p>Картофель - 4 штуки<br>
Морковь - 2 штуки<br>
Яйца - 4 штуки<br>
Колбаса - 300 грамм</p>
</body>
</html>
```

Заключение

Поздравляю! Вы только что создали свою первую полноценную веб-страницу!

Что Вы освоили в этом уроке:

- Создание правильно структуированного HTML-документа
- Использование заголовков разных уровней (<h1>, <h2>, <h3>)
- Форматирование текста с помощью абзацев (<p>)

- Добавление разделителей (<hr>)
- Сохранение и просмотр результата в браузере

Ваша страница-визитка теперь:

Имеет четкую структуру
Удобна для чтения
Корректно отображается в браузере
Может быть легко изменена и дополнена

Следующий шаг: Сохраните эту страницу и покажите её своим близким!
В следующих уроках мы добавим списки, картинки и ссылки, чтобы
сделать визитку еще более интересной.

Глава 5:

Простое оформление текста

5.1. Жирный шрифт **** или ****

Введение: Визуальное выделение vs Смысловое выделение

На первый взгляд, теги **** и **** делают одно и то же — делают текст жирным. Но между ними есть фундаментальная разница, которая важна для доступности, SEO и семантической разметки.

Часть 1: Базовое сравнение тегов

Тег **** (от "bold" — жирный)

html

```
<b>Этот текст будет жирным</b>
```

Результат: **Этот текст будет жирным**

Тег **** (от "strong" — сильный, важный)

html

```
<strong>Этот текст тоже будет жирным</strong>
```

Результат: **Этот текст тоже будет жирным**

Визуально они выглядят одинаково, но имеют разное значение!

Часть 2: Ключевые различия — аналогия

Давайте используем простую аналогию с речью человека:

**** — как повышение громкости

html

```
<p>Я купил <b>красные</b> яблоки и <b>зеленые</b> груши.</p>
```

| Я купил **красные** яблоки и **зеленые** груши.

Просто визуальное выделение, без изменения смысла

**** — как изменение интонации

html

<p>Внимание! Не забудьте полить растения вечером.</p>

Внимание! Не забудьте полить растения вечером.

Выделение важности, смены смыслового акцента

Часть 3: Технические различия

Характеристика		
Назначение	Визуальное выделение	Смысловое выделение важности
Семантика	Нет семантического значения	Имеет семантическое значение
Для скринридеров	Читается обычным тоном	Читается с измененной интонацией
SEO	Не влияет на поисковые системы	Может учитываться поисковиками
Визуальный эффект	Жирный шрифт	Жирный шрифт

Часть 4: Когда использовать — примеры

Пример 1: Выделение в тексте без изменения смысла

html

<p>В саду растут **красные** розы, **белые** лилии и **желтые** тюльпаны.</p>

Пример 2: Ключевые слова в описании

html

<p>Рецепт требует: **свежие** помидоры, **сладкий** перец и **молодой** картофель.</p>

Пример 3: Выделение терминов

html

<p>В HTML тег **** означает жирное начертание, а **<i>** — курсивное.</p>

Часть 5: Когда использовать `` — примеры

Пример 1: Предупреждения и важные уведомления

html

```
<p><strong>Важно!</strong> Не поливайте растения в солнечную погоду  
— это может вызвать ожоги листьев.</p>
```

Пример 2: Критическая информация

html

```
<p><strong>Внимание:</strong> Следующие действия нельзя отменить.  
Убедитесь в правильности выбора.</p>
```

Пример 3: Ключевые моменты инструкции

html

```
<p>Для посадки: <strong>выкопайте яму глубиной 50 см</strong>,  
добавьте удобрения, затем поместите саженец.</p>
```

Часть 6: Практические примеры с обоими тегами

Пример 1: Кулинарный рецепт

html

```
<p>Для приготовления салата Вам понадобятся:</p>  
<ul>  
<li><b>Свежие</b> огурцы — 2 штуки</li>  
<li><b>Спелые</b> помидоры — 3 штуки</li>  
<li><strong>Обязательно:</strong> соль и перец по вкусу</li>  
<li><strong>Важно:</strong> заправлять непосредственно перед  
подачей</li>  
</ul>
```

Пример 2: Инструкция по уходу за растениями

html

```
<p>Уход за розами включает:</p>  
<ul>  
<li><b>Регулярный</b> полив утром или вечером</li>  
<li><b>Систематическая</b> подкормка удобрениями</li>  
<li><strong>Обязательная</strong> обрезка весной</li>  
<li><strong>Внимание:</strong> защита от заморозков зимой</li>  
</ul>
```

Пример 3: Описание продукта

html

```
<p>Наш новый сорт томатов:</p>
<ul>
<li><b>Крупные</b> плоды весом до 300 грамм</li>
<li><b>Сочная</b> мякоть с насыщенным вкусом</li>
<li><strong>Особенность:</strong> устойчив к распространенным
заболеваниям</li>
<li><strong>Рекомендация:</strong> выращивать в защищенном
грунте</li>
</ul>
```

Часть 7: Как это "слышат" скринридеры⁵

Для слабовидящих пользователей:

С тегом :

text

"В саду растут красные розы, белые лилии и желтые тюльпаны."
Обычное чтение, без акцентов"

С тегом :

text

"Внимание! [пауза, изменение интонации] Не забудьте полить растения
вечером."

Изменение интонации, подчеркивание важности

Часть 8: Практическое задание

Создайте страницу с правильным использованием и :

html

```
<!DOCTYPE html>
<html lang="ru">
<head>
<title>Правильное выделение текста</title>
```

⁵ Скринридер (от англ. screen reader — "читатель экрана") — программа, которая преобразует текстовую информацию на экране в речь или в тактильные сигналы. Другие названия — программа экранного доступа или чтения, программа чтения с экрана, экранное считающее устройство.

```
<meta charset="utf-8">
</head>
<body>
<h1>Уход за комнатными растениями</h1>

<h2>Общие рекомендации</h2>
<p>Для <b>здорового</b> роста растений соблюдайте следующие правила:</p>

<h3>Полив</h3>
<p><b>Регулярный</b> полив — основа ухода. <strong>Важно:</strong> не допускайте пересыхания почвы.</p>

<h3>Освещение</h3>
<p>Большинство растений предпочитают <b>рассеянный</b> свет. <strong>Внимание:</strong> прямые солнечные лучи могут вызвать ожоги.</p>

<h3>Температура</h3>
<p>Поддерживайте <b>стабильную</b> температуру. <strong>Критично:</strong> избегайте резких перепадов.</p>

<h2>Частые ошибки</h2>
<p><strong>Самая распространенная ошибка:</strong> чрезмерный полив. <b>Избыточная</b> влага приводит к загниванию корней.</p>

<hr>

<p><strong>Запомните:</strong> правильный уход — залог <b>красивых</b> и <b>здоровых</b> растений!</p>
</body>
</html>
```

Часть 9: Советы по выбору тега

Используйте **** когда:

Нужно просто визуально выделить текст.
Выделение не меняет смысл предложения.
Это ключевые слова или термины.
Декоративное оформление.

Используйте **** когда:

Нужно подчеркнуть важность информации.

Это предупреждение или уведомление.

Критическая информация.

Ключевые моменты инструкции.

Простой чек-лист для выбора:

Это просто красиво? →

Это действительно важно? →

Скринридер должен это подчеркнуть? →

Это влияет на понимание смысла? →

Часть 10: Дополнительные возможности

Комбинирование тегов:

html

```
<p>Это <b><strong>очень важное</strong></b> выделение.</p>
```

Это **очень важное** выделение.

С CSS-стилизацией:

html

```
<style>
.warning {
font-weight: bold;
color: red;
}
</style>
```

```
<p><strong class="warning">Внимание! Опасность!</strong></p>
```

Часть 11: Частые ошибки

✗ НЕПРАВИЛЬНО:

html

```
<!-- Использование strong для простого выделения -->
```

```
<p>Купите <strong>красные</strong> яблоки и <strong>зеленые</strong>  
груши.</p>
```

```
<!-- Использование b для важных предупреждений -->
```

```
<p><b>Осторожно!</b> Высокое напряжение.</p>
```

ПРАВИЛЬНО:

html

```
<!-- b для визуального выделения -->  
<p>Купите <b>красные</b> яблоки и <b>зеленые</b> груши.</p>
```

```
<!-- strong для важных предупреждений -->
```

```
<p><strong>Осторожно!</strong> Высокое напряжение.</p>
```

Заключение

Краткие правила:

**** — для **визуального** выделения (просто жирный шрифт)

**** — для **смыслового** выделения (важная информация)

Что Вы теперь умеете:

-  Правильно выбирать между **** и ****
-  Понимать разницу в семантике
-  Создавать доступный контент для скринридеров
-  Улучшать SEO своих страниц

Запомните: Правильное использование этих тегов делает Ваш сайт не только красивым, но и понятным для всех пользователей, включая тех, кто использует вспомогательные технологии.

Теперь ваше выделение текста будет не только визуально привлекательным, но и семантически правильным!

5.2. Курсив <i> или

Введение: Визуальный акцент vs Смысловое ударение

Как и в случае с жирным шрифтом, теги <i> и визуально делают одно и то же — наклоняют текст. Но их семантическое значение различается так же фундаментально, как различаются просто "громкая речь" и "изменение интонации для передачи особого смысла".

Часть 1: Базовое сравнение тегов

Тег <i> (от "italic" — курсив)

html

```
<i>Этот текст будет курсивом</i>
```

Результат: *Этот текст будет курсивом*

Тег (от "emphasis" — акцент, ударение)

html

```
<em>Этот текст тоже будет курсивом</em>
```

Результат: *Этот текст тоже будет курсивом*

Визуально оба выглядят как курсив, но несут разную смысловую нагрузку!

Часть 2: Ключевые различия — аналогия

<i> — как изменение тембра голоса

html

```
<p>В романе <i>Война и мир</i> Толстой описывает...</p>
```

```
<p>Термин <i>фотосинтез</i> происходит от греческих слов...</p>
```

В романе *Война и мир* Толстой описывает...

Термин *фотосинтез* происходит от греческих слов...

Просто визуальное выделение, как бы "другим голосом"

 — как смысловое ударение

html

```
<p>Я <em>действительно</em> хочу научиться создавать сайты.</p>
```

```
<p>Это <em>самый</em> важный шаг в процессе посадки.</p>
```

Я действительно хочу научиться создавать сайты.
Это *самый* важный шаг в процессе посадки.

Изменение интонации для передачи особого смысла

Часть 3: Технические различия

Характеристика	<i>	
Назначение	Визуальное выделение	Смыслоное ударение
Семантика	Нет семантического значения	Имеет семантическое значение
Для скринридеров	Читается обычным тоном	Читается с измененной интонацией
SEO	Не влияет на поисковые системы	Может учитываться поисковиками
Визуальный эффект	Курсив	Курсив
Вложенность	Не усиливается при вложении	Усиливается при вложении

Часть 4: Когда использовать <i> — примеры

Пример 1: Названия произведений

html

```
<p>Я перечитываю <i>Преступление и наказание</i> Достоевского.</p>
<p>Смотрю фильм <i>Ирония судьбы, или С лёгким паром!</i></p>
```

Пример 2: Иностранные слова и термины

html

```
<p>Слово <i>компьютер</i> пришло из английского языка.</p>
<p>В ботанике используется термин <i>фотосинтез</i>. </p>
```

Пример 3: Научные названия

html

```
<p>Роза относится к виду <i>Rosa</i> семейства Розоцветные.</p>
<p>Картофель — <i>Solanum tuberosum</i>. </p>
```

Пример 4: Мысли, сны, мечты

html

<p><i>Интересно, получится ли у меня создать свой сайт?</i> — подумал я.</p>
<p>Ему приснилось: <i>я гуляю по цветущему саду</i>.</p>

Часть 5: Когда использовать — примеры

Пример 1: Смыслоное ударение

html
<p>Я обязательно научусь создавать сайты.</p>
<p>Это именно то, что мне нужно.</p>

Пример 2: Противопоставление

html
<p>Я хочу не просто сайт, а красивый и удобный сайт.</p>
<p>Поливать нужно утром, а не в полдень.</p>

Пример 3: Важные акценты в инструкциях

html
<p>Перед посадкой обязательно удобрите почву.</p>
<p>Ни в коем случае не поливайте растение днём на солнце.</p>

Часть 6: Практические примеры с обоими тегами

Пример 1: Литературный обзор

html
<p>Вчера я перечитывал <i>Евгения Онегина</i> Пушкина и действительно восхитился красотой языка. Главный герой — <i>Онегин</i> — показан как чрезвычайно сложная личность.</p>

Пример 2: Научная статья

html
<p>Процесс <i>фотосинтеза</i> — это основной способ питания растений.
Термин происходит от греческих слов <i>phōs</i> (свет) и <i>synthesis</i> (соединение). Важно понимать, что без этого процесса жизнь на Земле была бы невозможна.</p>

Пример 3: Личный дневник

html

```
<p><i>Сегодня такой прекрасный день!</i> — подумал я, выходя в сад.  
<em>Обязательно</em> нужно пересадить розы. Прочитал в книге  
<i>Секреты садоводства</i>, что это <em>лучше</em> делать  
весной.</p>
```

Часть 7: Особенность вложенности тегов

Тег *<i>* при вложении:

html

```
<p>Это <i>просто <i>курсив</i> внутри курсива</i>.</p>
```

Это просто курсив внутри курсива.

Визуальный эффект не усиливается

Тег ** при вложении:

html

```
<p>Это <em>очень <em>важный</em> акцент</em>.</p>
```

Это очень важный акцент.

Скринридер усиливает интонацию на вложенном элементе

Часть 8: Как это "слышат" скринридеры

Для слабовидящих пользователей:

С тегом *<i>*:

text

"Роман Война и мир Толстого считается классикой литературы."

Обычное чтение, без акцентов

С тегом **:

text

"Я [пауза, изменение интонации] действительно [возврат к обычной интонации] хочу научиться..."

Изменение интонации на выделенном слове

Часть 9: Практическое задание

Создайте страницу с правильным использованием *<i>* и **:

html

```
<!DOCTYPE html>
<html lang="ru">
<head>
<title>Курсив и смысловые акценты</title>
<meta charset="utf-8">
</head>
<body>
<h1>Мои литературные предпочтения</h1>

<h2>Любимые произведения</h2>
<p>Среди моих любимых книг — <i>Мастер и Маргарита</i> Булгакова и <i>Анна Каренина</i> Толстого. Я <em>действительно</em> считаю, что это величайшие произведения русской литературы.</p>

<h2>Чтение на иностранных языках</h2>
<p>Недавно я начал читать <i>Harry Potter</i> в оригинале. Слово <i>wand</i> означает "волшебная палочка". <em>Очень</em> рекомендую такой способ изучения языка!</p>

<h2>Ботанические интересы</h2>
<p>В саду я выращиваю <i>Rosa rugosa</i> (роза морщинистая) и <i>Paeonia lactiflora</i> (пион молочноцветковый). <em>Обязательно</em> нужно подкармливать их весной.</p>

<h2>Мои мысли</h2>
<p><i>Интересно, смогу ли я когда-нибудь создать свой собственный сайт о литературе и садоводстве?</i> — часто думаю я. Но я <em>точно</em> знаю, что буду стараться!</p>

<hr>

<p><em>Важно:</em> все названия произведений и научные термины выделены с помощью <i>&lt;i&gt;</i>, а смысловые акценты — с помощью <i>&lt;em&gt;</i>. </p>
</body>
</html>
```

Часть 10: Советы по выбору тега

Используйте **<i>** когда:

Названия книг, фильмов, картин

Иностранные слова и термины

Научные названия (латынь)

Мысли, сны, цитаты

Технические термины

Используйте **** когда:

Нужно сделать смысловое ударение

Противопоставление в предложении

Важные акценты в инструкциях

Эмоциональное выделение

Критически важная информация

Простой чек-лист для выбора:

Это название или термин? → <i>

Нужно изменить интонацию при чтении? →

Это просто оформление? → <i>

Это влияет на понимание смысла? →

Часть 11: Комбинирование с другими тегами

С жирным шрифтом:

html

```
<p>Это <b><i>очень важный термин</i></b> в ботанике.</p>
<p><strong><em>Внимание! Это критически важно!</em></strong></p>
```

В списках:

html

```
<ul>
<li><i>Rosa</i> — род растений семейства Розоцветные</li>
<li><em>Обязательно</em> поливайте утром или вечером</li>
<li>Книга <i>Сад для начинающих</i> — <em>лучший</em>
помощник</li>
</ul>
```

Часть 12: Историческая справка

HTML4 и ранее:

<*i*> — только визуальное оформление
<*em*> — только семантическое выделение

HTML5 (современный стандарт):

<*i*> — получил семантическое значение для конкретных случаев
<*em*> — сохранил семантику смыслового ударения

Часть 13: Частые ошибки

НЕПРАВИЛЬНО:

html
<!-- Использование *em* для названий -->
<p>Читаю <*em*>Войну и мир</*em*> Толстого.</p>

<!-- Использование *i* для смысловых акцентов -->
<p>Я <*i*>обязательно</*i*> приду завтра.</p>

<!-- Путаница с визуальным и смысловым -->
<p><*i*>Внимание!</*i*> Осторожно, скользко.</p>

ПРАВИЛЬНО:

html
<!-- *i* для названий -->
<p>Читаю <*i*>Войну и мир</*i*> Толстого.</p>

<!-- *em* для смысловых акцентов -->
<p>Я <*em*>обязательно</*em*> приду завтра.</p>

<!-- *strong* для важных предупреждений -->
<p><*strong*>Внимание!</*strong*> Осторожно, скользко.</p>

Заключение

Краткие правила:

<*i*> — для визуального и семантического выделения названий, терминов, мыслей
<*em*> — для смыслового ударения и изменения интонации

Что Вы теперь умеете:

- Правильно выбирать между <i> и .
- Понимать разницу в семантике и доступности.
- Создавать тексты, понятные скринридерам.
- Профессионально оформлять литературные и научные тексты.

Золотое правило:

Если выделение меняет **произношение** текста — используйте .

Если выделение меняет **категорию** текста (название, термин) — используйте <i>.

Теперь Ваши тексты будут не только красиво оформлены, но и семантически корректны, что особенно важно для образовательного и научного контента!

5.3. Почему есть два варианта? Разница между внешним видом и важностью.

Введение: Разделение ответственности — основа современного веб-дизайна

Существование пар тегов `/` и `<i>/` — это не случайность, а продуманное архитектурное решение. Давайте разберемся, почему разработчики HTML пошли по этому пути и какая философия стоит за этим разделением.

Часть 1: Исторический контекст — эволюция HTML

HTML 1.0-3.2: Эпоха визуального дизайна

html

```
<!-- Ранний HTML был ориентирован на внешний вид -->
<font size="5" color="red">Важный текст</font>
<b>Жирный текст</b>
<i>Курсивный текст</i>
<u>Подчеркнутый текст</u>
```

Проблема: Смешение структуры и оформления приводило к:

Недоступности для людей с ограниченными возможностями
Сложностям с поисковой оптимизацией
Трудностям в поддержке и изменении дизайна

HTML 4.0: Начало семантической революции

Введение тегов с **семантическим значением**:

`` вместо `` для важного контента
`` вместо `<i>` для акцентированного контента
`<cite>` для цитат и названий

HTML5: Баланс семантики и гибкости

Разработчики нашли компромисс:

`` и `<i>` получили новые семантические роли
`` и `` сохранили роль смыслового выделения

Часть 2: Философская разница — аналогии из реальной жизни

Аналогия 1: Дорожные знаки vs. Рекламные баннеры
``, `<i>` — как рекламные баннеры:

html

```
<p>Купите <b>красные</b> яблоки и <i>спелые</i> груши.</p>
```

Купите **красные** яблоки и *спелые* груши.

Просто привлекают внимание, не меняя правил

, — как дорожные знаки:

html

```
<p><strong>Стоп!</strong> Опасный поворот.</p>
<p>Поворачивайте <em>только</em> налево.</p>
```

Стоп! Опасный поворот.

Поворачивайте *только* налево.

Несут важную информацию, влияющую на поведение

Аналогия 2: Выделение маркером vs Подчеркивание ручкой
, <i> — выделение маркером:

html

```
<p>В рецепте нужны <b>свежие</b> помидоры и <i>молодой</i>
картофель.</p>
```

Просто визуальное выделение ключевых слов.

, — подчеркивание ручкой:

html

```
<p><strong>Внимание!</strong> Не добавляйте <em>ни в коем
случае</em> соль.</p>
```

Выделение, меняющее смысл и важность.

Часть 3: Техническая разница — как это работает "под капотом"
DOM-дерево и семантика:

html

```
<!-- Браузер создает такую структуру в памяти -->
<div>
<p>Это <b>просто жирный</b> текст</p>
<p>Это <strong>важный</strong> текст</p>
</div>
```

Для **:** Браузер видит просто текст с особым оформлением

Для **:** Браузер понимает, что этот текст имеет особую важность

Доступность (Accessibility):

Скринридеры (NVDA, JAWS, VoiceOver):

html

```
<p>Цена: <b>100 рублей</b>. <strong>Акция!</strong> Только
сегодня.</p>
```

Озвучивание:

"Цена: 100 рублей. [обычным тоном]"

"Акция! [измененной интонацией] Только сегодня."

Часть 4: Практические сценарии использования

Сценарий 1: Интернет-магазин

html

```
<div class="product">
<h3>Смартфон <i>SuperPhone X</i></h3>
<p>Цвет: <b>черный</b>, <b>белый</b>, <b>синий</b></p>
<p><strong>Осталось всего 3 штуки!</strong></p>
<p><em>Доставка только</em> по Москве</p>
</div>
```

Разбор:

<i> — название модели (семантика названия)
**** — варианты цвета (визуальное выделение)
**** — важная информация о количестве
**** — ограничение по доставке (смысловой акцент)

Сценарий 2: Медицинский сайт

html

```
<div class="instruction">
```

```
<h3>Прием лекарства <i>Aspirin</i></h3>
<p>Дозировка: <b>1 таблетка</b> 3 раза в день</p>
<p><strong>Противопоказания:</strong> беременность</p>
<p><em>Обязательно</em> проконсультируйтесь с врачом</p>
</div>
```

Часть 5: Влияние на поисковую оптимизацию (SEO)

Как поисковые системы видят разницу:

Google/Yandex для **/<i>**:

text

"Это просто оформление, не влияющее на релевантность"

Google/Yandex для **/**:

text

"Этот контент автор считает важным - учтем при ранжировании"

Практический SEO-пример:

```
html
<article>
<h1>Выращивание помидоров</h1>
<p>Сорт <i>Бычье сердце</i> требует особого ухода.</p>
<p><strong>Самая важная процедура</strong> — правильный полив.</p>
<p><em>Никогда</em> не поливайте в солнечную погоду.</p>
<p>Удобрения: <b>органические</b> и <b>минеральные</b>.</p>
</article>
```

Поисковая система поймет:

Выращивание помидоров

Сорт *Бычье сердце* требует особого ухода.

Самая важная процедура — правильный полив.

Никогда не поливайте в солнечную погоду.

Удобрения: органические и минеральные.

Часть 6: Будущее веб-стандартов — почему это важно

Тенденции развития:

Умные помощники (Алиса, Сири, Алекса)

Голосовой поиск

Автоматические переработки и суммаризация

Доступность как стандарт

Как семантические теги помогают:

html

```
<!-- Голосовой помощник понимает структуру -->
<article>
<p>Рецепт требует <b>спелые</b> помидоры.</p>
<p><strong>Критически важно:</strong> не использовать алюминиевую
посуду.</p>
<p>Блюдо <i>Рататуй</i> готовится 40 минут.</p>
</article>
```

Голосовой помощник может:

Выделить ключевые ингредиенты ("спелые помидоры")

Предупредить об опасности ("критически важно")

Правильно произнести название ("Рататуй")

Часть 7: Практическое упражнение — анализ и исправление

Давайте проанализируем и улучшим код:

Исходный код (с ошибками):

html

```
<p><i>Внимание!</i> Товар <b>очень</b> популярен.</p>
<p><em>iPhone 15</em> стоит <strong>99 000 рублей</strong>.</p>
<p>Цвета: <em>черный, белый, красный</em>.</p>
```

Исправленный код:

html

```
<p><strong>Внимание!</strong> Товар <em>очень</em> популярен.</p>
<p><i>iPhone 15</i> стоит <b>99 000 рублей</b>.</p>
<p>Цвета: <b>черный, белый, красный</b>.</p>
```

Объяснение исправлений:

"Внимание!" — важное предупреждение →
"очень" — смысловое ударение →
"iPhone 15" — название продукта → <i>
"99 000 рублей" — визуальное выделение цены →
Цвета — варианты характеристик →

Часть 8: Советы для правильного выбора

Простой алгоритм выбора:

Задайте вопрос: "Что я хочу передать?"

"Просто выделить визуально" → /<i>

"Подчеркнуть важность" → /

Проверка на "озвучивание":

Произнесите текст вслух

Если нужно изменить интонацию → /

Если просто выделить слово → /<i>

Проверка на доступность:

"Понятно ли будет скринридеру?"

"Нужно ли подчеркнуть важность для незрячих?"

Чек-лист для самопроверки:

Ситуация	Тег	Почему
Название книги	<i>	Семантика названия
Важное предупреждение		Критическая важность
Ключевое слово		Визуальное выделение
Смысловой акцент		Изменение интонации
Иностранные слова	<i>	Выделение иноязычного
Противопоставление		Смысловое ударение

Часть 9: Будущее развитие навыков

Что изучать дальше:

CSS — для полного контроля над внешним видом

ARIA-атрибуты — для улучшения доступности

Микроразметка — для лучшего понимания поисковиками

Веб-стандарты — чтобы быть в курсе изменений

Пример профессионального подхода:

html

```
<article itemscope itemtype="https://schema.org/Recipe">
<h1 itemprop="name">Салат <i>Оливье</i></h1>
<p><strong itemprop="description">Классический праздничный
салат</strong></p>
<div itemprop="ingredients">
<p>Ингредиенты: <b>картофель</b>, <b>морковь</b>,
<b>горошек</b></p>
</div>
<p><em itemprop="recipeInstructions">Важно:</em> нарезать мелкими
кубиками</p>
</article>
```

Заключение

Главные выводы:

Разделение визуального и семантического — основа современной вёрстки веб-сайтов.

, <i> — для оформления и специальных случаев (названия, термины)

**, ** — для передачи важности и смысловых акцентов

Правильный выбор улучшает доступность, SEO и пользовательский опыт

Почему это важно именно для вас:

Создаете доступные сайты для всех пользователей

Улучшаете позиции в поисковых системах

Делаете контент понятным для умных помощников

Следуете современным стандартам веб-разработки

Запомните: Хороший разработчик думает не только о том, как сайт выглядит, но и о том, как он "понимается" браузерами, поисковиками и вспомогательными технологиями.

Теперь Вы не просто делаете текст жирным или курсивным — Вы передаете смысл и важность!

5.4. Практика: выделяем важные мысли на своей визитке

Введение: Превращаем обычную визитку в эффективную

Теперь мы научимся правильно расставлять акценты на Вашей веб-визитке. Правильное выделение важных мыслей сделает Вашу страницу не только красивой, но и эффективной — посетители сразу увидят самое главное.

Часть 1: Подготовка — открываем нашу визитку

Найдите файл vizitka.html в папке "Мой первый сайт"

Откройте его в Блокноте (правой кнопкой → Открыть с помощью → Блокнот)

Создайте резервную копию (сохраните как vizitka_backup.html)

Часть 2: Анализ контента — что нужно выделить?

Давайте разберем нашу визитку и определим, какие элементы требуют особого внимания:

Какие элементы обычно выделяют:

- ✓ Ключевые навыки и компетенции
- ✓ Важные достижения и опыт
- ✓ Контактную информацию
- ✓ Призывы к действию
- ✓ Особые умения
- ✓ Уникальные предложения

Часть 3: Пошаговое улучшение визитки

Шаг 1: Добавляем структуру и заголовки

Было:

```
html
<h1>Добро пожаловать на мою страницу!</h1>
```

Стало:

```
html
<h1>Добро пожаловать на мою страницу!</h1>
```

<p>Меня зовут Александр Петров — начинающий веб-разработчик с большим энтузиазмом и желанием учиться.</p>

Шаг 2: Выделяем ключевые навыки

Было:

html

<h2>Немного о себе</h2>

<p>Здравствуйте! Меня зовут Александр, и я начинающий веб-разработчик.</p>

<p>Недавно вышел на пенсию и решил освоить новую для себя профессию...</p>

Стало:

html

<h2>Немного о себе</h2>

<p>Здравствуйте! Меня зовут Александр, и я начинающий веб-разработчик.</p>

<p>После 35 лет работы инженером решил освоить новую профессию — создание сайтов. Обладаю техническим складом ума и усидчивостью, что помогает в изучении программирования.</p>

Шаг 3: Акцентируем увлечения и хобби

Было:

html

<h2>Мои увлечения</h2>

<p>В свободное время я люблю работать в саду, читать книги и проводить время с внуками.</p>

<p>Особенно мне нравится выращивать розы...</p>

Стало:

html

<h2>Мои увлечения</h2>

<p>В свободное время я обожаю работать в саду, читать <i>исторические романы</i> и проводить время с внуками.</p>

<p>Особая страсть — выраживание роз. У меня коллекция из 15 сортов, включая <i>чайно-гибридные</i> и <i>плетистые⁶</i> разновидности.</p>

Шаг 4: Подчеркиваем мотивацию и цели

Было:

html

```
<h2>Почему я учу HTML</h2>
<p>Я верю, что никогда не поздно учиться чему-то новому...</p>
<p>Моя цель - создать собственный сайт о садоводстве...</p>
```

Стало:

html

```
<h2>Почему я учу HTML</h2>
<p>Я <strong>твердо верю</strong>, что <em>никогда не поздно</em>
учиться чему-то новому. Освоение веб-разработки помогает мне
<b>поддерживать умственную активность</b> и идти в ногу со
временем.</p>
<p><strong>Моя главная цель</strong> — создать <em>собственный
сайт о садоводстве</em>, где я смогу делиться своим <b>многолетним
опытом</b> с другими любителями растений.</p>
```

Часть 4: Полная улучшенная версия визитки

html

```
<!DOCTYPE html>
<html lang="ru">
<head>
<title>Александр Петров - Начинающий веб-разработчик</title>
<meta charset="utf-8">
</head>
<body>
<h1>Добро пожаловать на мою страницу!</h1>
<p>Меня зовут <strong>Александр Петров</strong> — <em>начинающий
веб-разработчик</em> с большим энтузиазмом и желанием учиться.</p>

<h2>Профессиональный путь</h2>
```

⁶ Плетистые розы — группа роз с длинными гибкими побегами, требующими опоры. Как правило, побеги покрыты шипами, а на концах стеблей и на боковых побегах образуются цветочные бутоны, одиночные или собранные в кисти.

<p>После 35 лет успешной работы инженером-проектировщиком принял решение освоить современную цифровую-профессию. Прошел самостоятельное обучение основам HTML и CSS, продолжаю углублять знания ежедневно.</p>

<p>Ключевые качества: аналитическое мышление, внимание к деталям, усидчивость и непрерывное стремление к развитию.</p>

<h2>Мои увлечения и хобби</h2>

<p>Страстный садовод с 20-летним стажем. Специализируюсь на выращивании <i>роз</i> и <i>плодовых кустарников</i>. Моя коллекция включает 15 уникальных сортов роз, включая редкие <i>английские сорта</i> Дэвида Остина.</p>

<p>Также увлекаюсь <i>исторической литературой</i> и цифровой фотографией природы. Провожу мастер-классы по садоводству для соседей и друзей.</p>

<h2>Почему я выбрал веб-разработку</h2>

<p>Искренне верю, что возраст — не помеха для освоения новых технологий. Веб-разработка позволяет сохранять ясность ума, развивать логическое мышление и быть частью современного мира.</p>

<p>Моя амбициозная цель — создание <i>интерактивного портала</i> для садоводов-любителей, где можно будет обмениваться опытом, вести дневники растений и находить единомышленников.</p>

<hr>

<h2>Контакты и сотрудничество</h2>

<p>Готов к взаимовыгодному сотрудничеству! Ищу наставника в веб-разработке и единомышленников для совместных проектов.</p>

<h3>Контактная информация</h3>

<p>Email: aleksandr.petrov@example.com</p>
<p>Телефон: +7 (123) 456-78-90</p>
<p>Город: <i>Москва</i></p>

<p>Открыт для новых знакомств и интересных предложений!</p>

</body>

</html>

Часть 5: Анализ наших улучшений

Что мы выделили и почему:

Элемент	Тег	Причина выделения
Имя		Главная идентификационная информация
Стаж работы		Важная количественная информация
Ключевые качества		Визуальное выделение списка качеств
Страстный садовод		Эмоциональная характеристика
Названия сортов	<i>	Специальные термины и названия
Коллекция роз		Важное достижение
Возраст — не помеха		Смысовой акцент на философии
Амбициозная цель		Ключевая мотивационная информация
Готов к сотрудничеству		Важный призыв к действию

Часть 6: Практическое задание — создайте свою уникальную визитку

Инструкция по созданию персонализированной визитки:

Создайте новый файл my_vizitka.html

Используйте шаблон ниже, заполнив его СВОИМИ данными:

```
html
<!DOCTYPE html>
<html lang="ru">
<head>
<title>[Ваше Имя] - [Ваша специализация]</title>
<meta charset="utf-8">
</head>
<body>
<h1>Добро пожаловать на мою персональную страницу!</h1>
```

<p>Меня зовут [Ваше Имя Фамилия] — [ваша краткая характеристика].</p>

<h2>Мой опыт и навыки</h2>

<p>[Ваши профессиональные навыки]. Обладаю [ключевое качество 1], [качество 2] и [самое важное качество]. </p>

<p>Прошел(ла) [обучение/курсы] и продолжаю развиваться в направлении <i>[ваша цель]</i>.</p>

<h2>Мои увлечения</h2>

<p>[Основное хобби] — это моя страсть уже [количество] лет. Особенно люблю <i>[конкретный аспект]</i> и <i>[другой аспект]</i>.</p>

<p>Также увлекаюсь [второстепенное хобби 1] и [второстепенное хобби 2].</p>

<h2>Мои цели в веб-разработке</h2>

<p>Главная цель — [ваша основная цель]. Верю, что [ваша философия или убеждение].</p>

<p>Планирую создать <i>[описание Вашего будущего проекта]</i>, который поможет [польза от проекта].</p>

<hr>

<h2>Контакты</h2>

<p>Email: [ваш email]</p>

<p>Телефон: [ваш телефон]</p>

<p>Город: <i>[ваш город]</i></p>

<p>[Ваш заключительный призыв или пожелание]</p>

</body>

</html>

Часть 7: Проверка и тестирование

Что проверить после создания:

Визуальная проверка в браузере:

Открыть файл в браузере.

Убедиться, что выделения заметны.

Проверить читабельность текста.

Семантическая проверка:

Прочитать текст вслух.
Убедиться, что акценты звучат естественно.
Проверить, не перегружен ли текст выделениями.

Техническая проверка:

Все теги закрыты правильно.
Русские буквы отображаются корректно.
Страница загружается без ошибок.

Чек-лист качества визитки:

Имя выделено **.**
Ключевые навыки выделены **или **.**
Особые увлечения выделены *.
Названия и термины выделены *.
Цели и мотивация подчеркнуты.
Контакты легко найти.
Текст не перегружен выделениями.
Общее впечатление профессиональное.****

Часть 8: Дополнительные улучшения

Добавьте стили для большего эффекта:

```
html
<head>
<title>Моя визитка</title>
<meta charset="utf-8">
<style>

strong {
    color: #2c3e50;
}

em {
    color: #e74c3c; font-style: italic;
}

b {
    color: #27ae60;
}

</style>
```

</head>

Добавьте разделители между секциями:

html

```
<hr style="border: 1px dashed #ccc; margin: 30px 0;">
```

```
<h2>Следующий раздел</h2>
```

Заключение

Что Вы достигли в этом уроке:

- Научились правильно расставлять акценты в тексте.**
- Поняли разницу между визуальным и смысловым выделением.**
- Создали профессиональную веб-визитку с правильной семантикой.**
- Научились анализировать контент на предмет важных элементов.**
- Освоили практическое применение тегов , , <i>, .**

Ваша визитка теперь:

Привлекает внимание к ключевой информации.

Легко читается и сканируется.

Производит профессиональное впечатление.

Доступна для скринридеров и поисковых систем.

Эффективно передает вашу уникальность и компетенции.

Следующий шаг: Покажите свою визитку друзьям и родным! Спросите их мнение — какие элементы привлекают внимание в первую очередь, что запоминается лучше всего. Это поможет Вам еще больше улучшить свои навыки веб-разработки!

Часть 3:

Создаём современную и удобную страницу

Глава 6: Списки — для порядка и структуры

6.1. Маркированный список и (для перечня чего угодно)

Введение: Организуем информацию с помощью списков

Представьте, что Вам нужно записать список продуктов для покупки или перечислить Ваши увлечения. Если просто написать их в строку через запятую, читать будет неудобно. Маркированные списки решают эту проблему — они делают любые перечни наглядными, структуризованными и легко читаемыми.

Часть 1: Что такое маркированный список?

Маркированный список — это список элементов, где каждый элемент помечен специальным символом (маркером), обычно точкой, кружком или квадратиком.

Базовая структура:

```
html
<ul>
<li>Первый элемент списка</li>
<li>Второй элемент списка</li>
<li>Третий элемент списка</li>
</ul>
```

Результат:

- Первый элемент списка
- Второй элемент списка
- Третий элемент списка

Часть 2: Теги маркированного списка и их назначение

 (Unordered List — неупорядоченный список)

Главный контейнер для всего списка.

Обозначает, что порядок элементов **не важен**.

По умолчанию добавляет отступы слева.

 (List Item — элемент списка)

Каждый отдельный пункт списка.
Должен находиться **внутри** или .
Автоматически начинается с новой строки.

Часть 3: Простые примеры использования

Пример 1: Список покупок

```
html
<h3>Список продуктов</h3>
<ul>
<li>Молоко</li>
<li>Хлеб</li>
<li>Яйца</li>
<li>Сыр</li>
<li>Фрукты</li>
</ul>
```

Список продуктов

- Молоко
- Хлеб
- Яйца
- Сыр
- Фрукты

Пример 2: Список увлечений

```
html
<h3>Мои хобби</h3>
<ul>
<li>Садоводство</li>
<li>Чтение книг</li>
<li>Фотография</li>
<li>Вязание</li>
<li>Приготовление выпечки</li>
</ul>
```

Пример 3: Список навыков

```
html
```

```
<h3>Что я умею в HTML</h3>
<ul>
<li>Создавать заголовки</li>
<li>Форматировать текст</li>
<li>Добавлять абзацы</li>
<li>Создавать списки</li>
<li>Добавлять изображения</li>
</ul>
```

Часть 4: Многоуровневые (вложенные) списки

Вы можете создавать списки внутри списков для сложной структуры:

Пример: Планы на неделю

```
html
<h3>Мои планы на неделю</h3>
<ul>
<li>Понедельник
<ul>
<li>Купить продукты</li>
<li>Полить цветы</li>
</ul>
</li>
<li>Вторник
<ul>
<li>Урок HTML</li>
<li>Приготовить обед</li>
</ul>
</li>
<li>Среда
<ul>
<li>Встреча с друзьями</li>
<li>Чтение книги</li>
</ul>
</li>
</ul>
```

Мои планы на неделю

- Понедельник
 - Купить продукты
 - Полить цветы
- Вторник
 - Урок HTML
 - Приготовить обед
- Среда
 - Встреча с друзьями
 - Чтение книги

Важно: Каждый вложенный список помещается **внутрь** тега родительского списка.

Часть 5: Комбинирование списков с другими тегами

Пример 1: Списки с выделением текста

```
html
<h3>Рецепт салата</h3>
<ul>
<li><strong>Помидоры</strong> — 2 штуки</li>
<li><em>Свежие</em> огурцы — 3 штуки</li>
<li>Лук <i>красный</i> — 1 головка</li>
<li><b>Соль</b> и перец по вкусу</li>
</ul>
```

Рецепт салата

- **Помидоры** — 2 штуки
- *Свежие* огурцы — 3 штуки
- Лук *красный* — 1 головка
- **Соль** и перец по вкусу

Пример 2: Список с ссылками

```
html
<h3>Полезные сайты для учебы</h3>
<ul>
```

```
<li><a href="https://htmlbook.ru">HTML Book</a> — справочник по  
HTML</li>  
<li><a href="https://validator.w3.org">W3C Validator</a> — проверка  
кода</li>  
<li><a href="https://ru.wikipedia.org">Википедия</a> — энциклопедия</li>  
</ul>
```

Полезные сайты для учебы

- [HTML Book](#) — справочник по HTML
- [W3C Validator](#) — проверка кода
- [Википедия](#) — энциклопедия

Часть 6: Практическое задание

Создайте страницу "Мои интересы" с использованием списков:

```
html  
<!DOCTYPE html>  
<html lang="ru">  
<head>  
<title>Мои интересы и увлечения</title>  
<meta charset="utf-8">  
</head>  
<body>  
<h1>Мир моих увлечений</h1>  
  
<h2>Любимые занятия</h2>  
<ul>  
<li><strong>Садоводство</strong> — выращиваю цветы и овощи</li>  
<li><em>Кулинария</em> — люблю готовить новые блюда</li>  
<li><b>Чтение</b> — предпочитаю исторические романы</li>  
<li>Фотография — снимаю природу и семью</li>  
</ul>  
  
<h2>Что выращиваю в саду</h2>  
<ul>  
<li>Цветы  
<ul>  
<li>Розы  
<ul>  
<li>Чайно-гибридные</li>
```

```
<li>Плетистые</li>
<li>Почвопокровные</li>
</ul>
</li>
<li>Пионы</li>
<li>Тюльпаны</li>
</ul>
</li>
<li>Овощи
<ul>
<li>Помидоры</li>
<li>Огурцы</li>
<li>Картофель</li>
</ul>
</li>
<li>Зелень
<ul>
<li>Укроп</li>
<li>Петрушка</li>
<li>Базилик</li>
</ul>
</li>
</ul>

<h2>Мои цели в изучении HTML</h2>
<ul>
<li>Научиться создавать красивые сайты</li>
<li>Сделать персональный блог о садоводстве</li>
<li>Помогать друзьям с их сайтами</li>
<li>Не отставать от современных технологий</li>
</ul>
</body>
</html>
```

Мир моих увлечений

Любимые занятия

- Садоводство — выращиваю цветы и овощи
- Кулинария — люблю готовить новые блюда
- Чтение — предпочитаю исторические романы
- Фотография — снимаю природу и семью

Что выращиваю в саду

- Цветы
 - Розы
 - Чайно-гибридные
 - Плетистые
 - Почвопокровные
 - Пионы
 - Тюльпаны
- Овощи
 - Помидоры
 - Огурцы
 - Картофель
- Зелень
 - Укроп
 - Петрушка
 - Базилик

Мои цели в изучении HTML

- Научиться создавать красивые сайты
- Сделать полнофункциональный блог с авторизацией

Часть 7: Типы маркеров (CSS-стилизация)

Хотя в чистом HTML нельзя изменить маркеры, с помощью CSS можно использовать разные стили:

Круглые маркеры (по умолчанию):

```
html
<ul style="list-style-type: disc;">
<li>Элемент с круглым маркером</li>
</ul>
```

Квадратные маркеры:

```
html
<ul style="list-style-type: square;">
<li>Элемент с квадратным маркером</li>
</ul>
```

Пустые кружки:

```
html
<ul style="list-style-type: circle;">
<li>Элемент с пустым маркером</li>
</ul>
```

Без маркеров:

```
html
<ul style="list-style-type: none;">
<li>Элемент без маркера7</li>
</ul>
```

Часть 8: Частые ошибки и как их избежать

✖ НЕПРАВИЛЬНО:

```
html
<!-- Тег li вне ul -->
<li>Элемент без списка</li>

<!-- Неправильная вложенность -->
<ul>
<li>Элемент</li>
<ul> <!-- Должен быть внутри li -->
<li>Подэлемент</li>
</ul>
</ul>

<!-- Забыли закрыть теги -->
<ul>
```

⁷ Используется при создании меню навигации на веб-сайте.

```
<li>Первый элемент  
<li>Второй элемент  
</ul>
```

ПРАВИЛЬНО:

```
html  
!-- Правильная структура -->  
<ul>  
<li>Первый элемент</li>  
<li>Второй элемент</li>  
</ul>  
  
!-- Правильная вложенность -->  
<ul>  
<li>Элемент  
<ul>  
<li>Подэлемент</li>  
</ul>  
</li>  
</ul>
```

Часть 9: Семантика и доступность

Для скринридеров:

Когда скринридер встречает список, он сообщает:

"Список из 5 элементов"

После каждого элемента: "элемент списка"

При завершении: "конец списка"

Для поисковых систем:

Списки помогают поисковикам понимать структуру контента и выделять ключевые элементы.

Часть 10: Дополнительные возможности

Список с описаниями:

```
html  
<ul>  
<li>  
<strong>HTML</strong> — язык разметки для создания веб-страниц  
</li>
```

```
<li>
<strong>CSS</strong> — язык стилей для оформления веб-страниц
</li>
</ul>
```

Список с иконками (с помощью CSS):

```
html
<ul style="list-style-type: '✓';">
<li>Выполненная задача</li>
<li>Еще одна задача</li>
</ul>
```

- ✓ Выполненная задача
- ✓ Еще одна задача

Заключение

Что Вы теперь умеете:

- Создавать простые маркированные списки
- Строить многоуровневые вложенные списки
- Комбинировать списки с другими HTML-тегами
- Избегать распространенных ошибок
- Понимать семантику списков

Преимущества использования списков:

-  Улучшают читаемость — информация воспринимается легче
-  Структурируют контент — помогают организовать мысли
-  Улучшают доступность — помогают скринридерам
-  Повышают SEO — поисковики лучше понимают структуру

Когда использовать маркированные списки:

Перечни объектов, предметов, понятий

Списки без определенного порядка

Перечисления характеристик

Любые наборы однородных элементов

Теперь Вы можете превращать скучные текстовые перечисления в красивые, структурированные списки, которые легко читать и понимать!

6.2. Нумерованный список и (для инструкций или рецептов)

Введение: Порядок имеет значение!

Когда последовательность действий важна, на помощь приходят нумерованные списки. В отличие от маркированных списков, где порядок не важен, нумерованные списки показывают четкую последовательность шагов — как в рецептах, инструкциях или руководствах.

Часть 1: Что такое нумерованный список?

Нумерованный список — это список элементов, где каждый элемент пронумерован по порядку. Браузер автоматически присваивает номера, которые обновляются при добавлении или удалении элементов.

Базовая структура:

```
html
<ol>
<li>Первый шаг</li>
<li>Второй шаг</li>
<li>Третий шаг</li>
</ol>
```

Результат:

- 1. Первый шаг
- 2. Второй шаг
- 3. Третий шаг

Часть 2: Теги нумерованного списка и их назначение

 (Ordered List — упорядоченный список)

Главный контейнер для нумерованного списка.
Обозначает, что порядок элементов **важен**.
Автоматически нумерует элементы (1, 2, 3...).

 (List Item — элемент списка)

Каждый отдельный шаг или пункт.
Работает так же, как в маркированных списках.

Номер присваивается автоматически.

Часть 3: Простые примеры использования

Пример 1: Рецепт приготовления чая

```
html
<h3>Приготовление чая</h3>
<ol>
<li>Вскипятите воду</li>
<li>Обдайте чайник кипятком</li>
<li>Положите заварку в чайник</li>
<li>Залейте кипятком</li>
<li>Настаивайте 5-7 минут</li>
</ol>
```

Приготовление чая

1. Вскипятите воду
2. Обдайте чайник кипятком
3. Положите заварку в чайник
4. Залейте кипятком
5. Настаивайте 5-7 минут

Пример 2: Инструкция по посадке растений

```
html
<h3>Посадка розы</h3>
<ol>
<li>Выкопайте яму 50×50 см</li>
<li>На дно насыпьте дренаж</li>
<li>Добавьте удобрения</li>
<li>Поместите саженец в яму</li>
<li>Засыпьте землей и уплотните</li>
<li>Обильно полейте</li>
</ol>
```

Пример 3: Утренние мероприятия

```
html
<h3>Мое утро</h3>
<ol>
```

```
<li>Проснуться в 7:00</li>
<li>Сделать зарядку</li>
<li>Принять душ</li>
<li>Позавтракать</li>
<li>Проверить почту</li>
<li>Начать работу над HTML</li>
</ol>
```

Часть 4: Атрибуты нумерованного списка

start — начать с определенного номера

html

```
<ol start="5">
<li>Пятый шаг</li>
<li>Шестой шаг</li>
<li>Седьмой шаг</li>
</ol>
```

- 5. Пятый шаг
- 6. Шестой шаг
- 7. Седьмой шаг

reversed — обратная нумерация

html

```
<ol reversed>
<li>Третий шаг</li>
<li>Второй шаг</li>
<li>Первый шаг</li>
</ol>
```

- 3. Третий шаг
- 2. Второй шаг
- 1. Первый шаг

type — тип нумерации

html

```
<!-- Арабские цифры (по умолчанию) -->
<ol type="1">
```

```

<li>Первый</li>
<li>Второй</li>
</ol>

<!-- Заглавные буквы --&gt;
&lt;ol type="A"&gt;
&lt;li&gt;Первый&lt;/li&gt;
&lt;li&gt;Второй&lt;/li&gt;
&lt;/ol&gt;

<!-- Строчные буквы --&gt;
&lt;ol type="a"&gt;
&lt;li&gt;Первый&lt;/li&gt;
&lt;li&gt;Второй&lt;/li&gt;
&lt;/ol&gt;

<!-- Римские цифры (заглавные) --&gt;
&lt;ol type="I"&gt;
&lt;li&gt;Первый&lt;/li&gt;
&lt;li&gt;Второй&lt;/li&gt;
&lt;/ol&gt;

<!-- Римские цифры (строчные) --&gt;
&lt;ol type="i"&gt;
&lt;li&gt;Первый&lt;/li&gt;
&lt;li&gt;Второй&lt;/li&gt;
&lt;/ol&gt;
</pre>

```

Часть 5: Многоуровневые нумерованные списки

Пример: Сложный рецепт с этапами

```

html
<h3>Приготовление торта "Наполеон"</h3>
<ol>
<li>Приготовление теста
<ol>
<li>Смешать муку и масло</li>
<li>Добавить холодную воду</li>
<li>Замесить тесто</li>
<li>Охладить 30 минут</li>
</ol>
</li>
<li>Приготовление крема
<ol>

```

```
<li>Взбить яйца с сахаром</li>
<li>Добавить молоко</li>
<li>Варить на медленном огне</li>
<li>Охладить</li>
</ol>
</li>
<li>Сборка торта
<ol>
<li>Раскатать коржи</li>
<li>Испечь до золотистого цвета</li>
<li>Промазать кремом каждый корж</li>
<li>Украсить торт</li>
</ol>
</li>
</ol>
```

Часть 6: Комбинирование с другими тегами

Пример 1: Рецепт с выделениями

```
html
<h3>Рецепт омлета</h3>
<ol>
<li><strong>Разбейте</strong> 3 яйца в миску</li>
<li><em>Взбейте</em> их венчиком</li>
<li>Добавьте <b>соль</b> и <b>перец</b></li>
<li>Нагрейте сковороду с <i>маслом</i></li>
<li>Вылейте смесь и жарьте 5 минут</li>
</ol>
```

Пример 2: Инструкция с временными рамками

```
html
<h3>Уход за рассадой помидоров</h3>
<ol>
<li>Полив — <strong>каждые 2 дня</strong></li>
<li>Подкормка — <em>раз в неделю</em></li>
<li>Проветривание — <b>ежедневно</b></li>
<li>Пересадка — через <i>30 дней</i></li>
</ol>
```

Часть 7: Практическое задание

Создайте страницу с любимым рецептом:

```
html
<!DOCTYPE html>
<html lang="ru">
<head>
<title>Мой фирменный рецепт</title>
<meta charset="utf-8">
</head>
<body>
<h1>Блины классические</h1>

<h2>Ингредиенты:</h2>
<ul>
<li>Молоко — 500 мл</li>
<li>Яйца — 3 шт.</li>
<li>Мука — 200 г</li>
<li>Сахар — 2 ст. ложки</li>
<li>Соль — ½ ч. ложки</li>
<li>Масло растительное — 2 ст. ложки</li>
</ul>

<h2>Приготовление:</h2>
<ol>
<li><strong>Подготовьте продукты:</strong> все ингредиенты должны быть комнатной температуры</li>
<li><em>Взбейте яйца</em> с сахаром и солью до легкой пены</li>
<li>Постепенно добавьте <b>половину молока</b>, продолжая взбивать</li>
<li>Просейте муку и добавьте в смесь, тщательно размешайте</li>
<li>Влейте оставшееся молоко и растительное масло</li>
<li>Дайте тесту <i>постоять 20-30 минут</i></li>
<li>Разогрейте сковороду и смажьте маслом</li>
<li>Выливайте тесто половником и жарьте с двух сторон до золотистой корочки</li>
</ol>

<h2>Советы:</h2>
<ul>
<li>Первый блин всегда получается комом — это нормально!</li>
<li>Для тонких блинов используйте больше молока</li>
<li>Подавайте с вареньем, сметаной или медом</li>
</ul>
</body>
</html>
```

Часть 8: Смешанные списки — комбинация ul и ol

Пример: План обучения HTML

```
html
<h3>Мой план изучения HTML</h3>
<ol>
<li>Базовые теги
<ul>
<li>Заголовки h1-h6</li>
<li>Абзацы p</li>
<li>Списки ul, ol, li</li>
</ul>
</li>
<li>Форматирование текста
<ul>
<li>Жирный текст b, strong</li>
<li>Курсив i, em</li>
<li>Подчеркивание</li>
</ul>
</li>
<li>Ссылки и изображения
<ul>
<li>Тег a для ссылок</li>
<li>Тег img для картинок</li>
<li>Атрибуты href и src</li>
</ul>
</li>
</ol>
```

Часть 9: Семантика и доступность

Для скринридеров:

Когда скринридер встречает нумерованный список, он сообщает:
"Нумерованный список из 6 элементов"

"Шаг 1: [текст]"

"Шаг 2: [текст]"

При завершении: "конец списка"

Для пользователей:

Четкая последовательность действий
Легко отслеживать прогресс

Просто вернуться к конкретному шагу

Часть 10: Частые ошибки и лучшие практики

✗ НЕПРАВИЛЬНО:

```
html
<!-- Неправильная вложенность -->
<ol>
<li>Первый шаг</li>
<ol> <!-- Должен быть внутри li -->
<li>Подшаг</li>
</ol>
</ol>

<!-- Ручная нумерация -->
<ol>
<li>1. Первый шаг</li> <!-- Не нужно! Браузер сам пронумерует -->
<li>2. Второй шаг</li>
</ol>

<!-- Смешивание типов списков неправильно -->
<ol>
<li>Шаг с маркированным списком:
• Подпункт <!-- Используйте теги! -->
• Подпункт
</li>
</ol>
```

✓ ПРАВИЛЬНО:

```
html
<!-- Правильная вложенность -->
<ol>
<li>Первый шаг
<ol>
<li>Подшаг</li>
</ol>
</li>
</ol>

<!-- Автоматическая нумерация -->
<ol>
<li>Первый шаг</li> <!-- Браузер сам поставит "1." -->
<li>Второй шаг</li> <!-- Браузер сам поставит "2." -->
```

```
</ol>

<!-- Правильное смешивание -->
<ol>
<li>Шаг с маркированным списком:
<ul>
<li>Подпункт</li>
<li>Подпункт</li>
</ul>
</li>
</ol>
```

Часть 11: Когда использовать нумерованные списки

Идеальные сценарии:

-  **Рецепты** — последовательность приготовления
-  **Инструкции** — пошаговые руководства
-  **Чек-листы** — списки действий по порядку
-  **Учебные материалы** — этапы обучения
-  **Планы** — расписания и графики

Пример из жизни:

```
html
<h3>Как посадить дерево</h3>
<ol>
<li>Выберите подходящее место</li>
<li>Выкопайте яму в 2 раза шире корней</li>
<li>Добавьте компост на дно ямы</li>
<li>Поместите дерево в яму</li>
<li>Засыпьте землей и уплотните</li>
<li>Обильно полейте</li>
<li>Замульчируйте приствольный круг</li>
</ol>
```

Заключение

Что Вы теперь умеете:

-  Создавать нумерованные списки для последовательных действий
-  Использовать атрибуты `start`, `reversed`, `type`
-  Строить многоуровневые нумерованные списки
-  Комбинировать нумерованные и маркированные списки
-  Создавать понятные инструкции и рецепты

Ключевые преимущества:



- Автоматическая нумерация** — не нужно нумеровать вручную
- Четкая последовательность** — идеально для инструкций
- Улучшенная доступность** — скринридеры понимают структуру
- Легкое редактирование** — добавили шаг, номера обновились автоматически

Запомните:

Используйте `` когда **порядок важен**

Используйте `` когда порядок **не имеет значения**

Всегда используйте правильную вложенность тегов

Теперь Вы можете создавать профессионально выглядящие инструкции, рецепты и руководства, которые будут понятны и удобны для всех пользователей!

6.3. Практика: добавляем на страницу список своих увлечений или список дел на день

Введение: Превращаем теорию в практику!

Пришло время применить полученные знания на практике. Мы создадим две полезные страницы: одну с Вашиими увлечениями (используя маркированные списки), другую с планами на день (используя нумерованные списки). Это поможет закрепить разницу между этими типами списков.

Часть 1: Подготовка рабочего пространства

Создаем новые файлы:

Откройте папку "Мой первый сайт"

Создайте два новых файла:

hobbies.html — для списка увлечений

todo.html — для списка дел на день

Базовая структура для обоих файлов:

```
html
<!DOCTYPE html>
<html lang="ru">
<head>
<title>Заголовок страницы</title>
<meta charset="utf-8">
</head>
<body>
<!-- Здесь будет наш контент --&gt;
&lt;/body&gt;
&lt;/html&gt;</pre>
```

Часть 2: Создаем страницу с увлечениями (hobbies.html)

Шаг 1: Добавляем заголовок и введение

```
html
<body>
<h1>Мои увлечения и хобби</h1>
<p>В свободное время я люблю заниматься разными интересными делами. Вот мои основные увлечения:</p>
```

Шаг 2: Создаем основной список увлечений

html

```
<h2>Мои основные хобби</h2>
<ul>
<li><strong>Садоводство</strong> — выращиваю цветы и овощи на
даче</li>
<li><em>Чтение книг</em> — особенно люблю исторические романы</li>
<li><b>Кулинария</b> — экспериментирую с новыми рецептами</li>
<li>Фотография — снимаю природу и семейные мероприятия</li>
<li>Рукоделие — вязание и вышивание</li>
</ul>
```

Шаг 3: Добавляем вложенные списки для детализации

html

```
<h2>Подробнее о моих увлечениях</h2>
```

```
<h3>Садоводство</h3>
<ul>
<li>Цветы
<ul>
<li>Розы (15 сортов)</li>
<li>Пионы</li>
<li>Тюльпаны и нарциссы</li>
</ul>
</li>
<li>Овощи
<ul>
<li>Помидоры</li>
<li>Огурцы</li>
<li>Зелень (петрушка, укроп, базилик)</li>
</ul>
</li>
</ul>
```

```
<h3>Чтение</h3>
<ul>
<li>Любимые жанры:
<ul>
<li>Исторические романы</li>
<li>Биографии</li>
<li>Детективы</li>
</ul>
```

```
</li>
<li>Любимые авторы:
<ul>
<li>Александр Дюма</li>
<li>Андрей Дементьев</li>
<li>Эдуард Асадов</li>
</ul>
</li>
</ul>
```

Шаг 4: Добавляем список желаний (будущих увлечений)

```
html
<h2>Чему я хочу научиться в будущем</h2>
<ul>
<li>Освоить веб-дизайн</li>
<li>Научиться играть на фортепиано</li>
<li>Изучить английский язык</li>
<li>Попробовать гончарное дело</li>
<li>Научиться рисовать акварелью</li>
</ul>
```

Часть 3: Полный код страницы hobbies.html

```
html
<!DOCTYPE html>
<html lang="ru">
<head>
<title>Мои увлечения</title>
<meta charset="utf-8">
</head>
<body>
<h1>Мои увлечения и хобби</h1>
<p>В свободное время я люблю заниматься разными интересными делами. Вот мои основные увлечения:</p>

<h2>Мои основные хобби</h2>
<ul>
<li><strong>Садоводство</strong> — выращиваю цветы и овощи на даче</li>
<li><em>Чтение книг</em> — особенно люблю исторические романы</li>
<li><b>Кулинария</b> — экспериментирую с новыми рецептами</li>
<li>Фотография — снимаю природу и семейные мероприятия</li>
<li>Рукоделие — вязание и вышивание</li>
<li>Изучение HTML — создаю свои первые веб-страницы</li>
```


<h2>Подробнее о моих увлечениях</h2>

<h3>Садоводство</h3>

Цветы

Розы (15 сортов)

Пионы

Тюльпаны и нарциссы

Георгины

Овощи

Помидоры

Огурцы

Картофель

Зелень (петрушка, укроп, базилик)

Фрукты и ягоды

Клубника

Смородина

Яблоки

<h3>Чтение</h3>

Любимые жанры:

Исторические романы

Биографии известных людей

Детективы

Классическая литература

Любимые авторы:

Александр Дюма

Андрей Дементьев

```
<li>Эдуард Асадов</li>
<li>Лев Толстой</li>
</ul>
</li>
<li>Последние прочитанные книги:
<ul>
<li>"Три мушкетера" — Александр Дюма</li>
<li>"Война и мир" — Лев Толстой</li>
</ul>
</li>
</ul>
```

```
<h3>Кулинария</h3>
<ul>
<li>Любимые блюда для приготовления:
<ul>
<li>Домашняя выпечка</li>
<li>Суп-пюре из овощей</li>
<li>Салаты по сезону</li>
<li>Домашнее варенье</li>
</ul>
</li>
<li>Специализация:
<ul>
<li>Блины и оладьи</li>
<li>Пироги с разными начинками</li>
<li>Заготовки на зиму</li>
</ul>
</li>
</ul>
```

```
<h2>Чему я хочу научиться в будущем</h2>
<ul>
<li>Освоить веб-дизайн и CSS</li>
<li>Научиться играть на фортепиано</li>
<li>Изучить английский язык</li>
<li>Попробовать гончарное дело</li>
<li>Научиться рисовать акварелью</li>
<li>Освоить цифровую фотографию</li>
</ul>
```

```
<hr>
<p><em>Обновлено: сегодняшняя дата</em></p>
</body>
</html>
```

Часть 4: Создаем страницу со списком дел (todo.html)

Шаг 1: Добавляем заголовок и введение

```
html
<body>
<h1>Мой план на сегодня</h1>
<p>Вот что я планирую сделать сегодня. Порядок выполнения важен!
</p>
```

Шаг 2: Создаем основной список дел

```
html
<h2>Утренние дела (с 8:00 до 12:00)</h2>
<ol>
<li><strong>Проснуться</strong> в 8:00</li>
<li><em>Сделать</em> утреннюю зарядку</li>
<li>Принять контрастный душ</li>
<li>Приготовить и съесть завтрак</li>
<li>Проверить электронную почту</li>
</ol>
```

Шаг 3: Добавляем дневные задачи

```
html
<h2>Дневные задачи (с 12:00 до 17:00)</h2>
<ol start="6">
<li>Заняться изучением HTML
<ol type="a">
<li>Повторить пройденные теги</li>
<li>Выполнить практическое задание</li>
<li>Создать новую страницу</li>
</ol>
</li>
<li>Сделать перерыв на обед</li>
<li>Работа в саду
<ol type="a">
<li>Полить цветы</li>
<li>Прополоть грядки</li>
<li>Подвязать помидоры</li>
</ol>
</li>
</ol>
```

Шаг 4: Добавляем вечерние занятия

```
html
<h2>Вечернее время (с 17:00 до 22:00)</h2>
<ol start="9">
<li>Приготовить ужин</li>
<li>Пообщаться с семьей</li>
<li>Почитать книгу 30 минут</li>
<li>Посмотреть интересный фильм</li>
<li>Подвести итоги дня</li>
<li>Подготовиться к завтрашнему дню</li>
</ol>
```

Часть 5: Полный код страницы todo.html

```
html
<!DOCTYPE html>
<html lang="ru">
<head>
<title>Мой план на день</title>
<meta charset="utf-8">
</head>
<body>
<h1>Мой план на сегодня</h1>
<p>Вот что я планирую сделать сегодня. Порядок выполнения важен!
</p>

<h2>Утренние дела (с 8:00 до 12:00)</h2>
<ol>
<li><strong>Проснуться</strong> в 8:00</li>
<li><em>Сделать</em> утреннюю зарядку (15 минут)</li>
<li>Принять контрастный душ</li>
<li>Приготовить и съесть питательный завтрак
<ul>
<li>Овсяная каша с фруктами</li>
<li>Чай с медом</li>
<li>Бутерброд с сыром</li>
</ul>
</li>
<li>Проверить электронную почту и сообщения</li>
<li>Запланировать задачи на день</li>
</ol>

<h2>Дневные задачи (с 12:00 до 17:00)</h2>
```

```
<ol start="7">
<li>Заняться изучением HTML (2 часа)
<ol type="a">
<li>Повторить пройденные темы (30 минут)</li>
<li>Выполнить практическое задание (1 час)</li>
<li>Создать новую страницу (30 минут)</li>
</ol>
</li>
<li>Сделать перерыв на обед (45 минут)
<ul>
<li>Овощной суп</li>
<li>Салат из свежих овощей</li>
<li>Компот</li>
</ul>
</li>
<li>Работа в саду (1.5 часа)
<ol type="a">
<li>Полить цветы и овощи</li>
<li>Прополоть грядки с морковью</li>
<li>Подвязать помидоры</li>
<li>Удалить сухие листья с роз</li>
</ol>
</li>
<li>Отдых и перерыв (30 минут)</li>
</ol>

<h2>Вечернее время (с 17:00 до 22:00)</h2>
<ol start="11">
<li>Приготовить ужин для семьи
<ul>
<li>Запеченная рыба с овощами</li>
<li>Гречневая каша</li>
<li>Свежий салат</li>
</ul>
</li>
<li>Семейный ужин и общение (1 час)</li>
<li>Чтение книги (30 минут)
<ul>
<li>Текущая книга: "Анна Каренина"</li>
<li>Цель: прочитать 20 страниц</li>
</ul>
</li>
<li>Прогулка на свежем воздухе (30 минут)</li>
<li>Посмотреть интересный фильм или сериал (1.5 часа)</li>
<li>Подвести итоги дня и записать в дневник</li>
```

```
<li>Подготовиться к завтрашнему дню
<ol type="a">
<li>Составить список задач на завтра</li>
<li>Подготовить одежду</li>
<li>Проверить календарь на завтра</li>
</ol>
</li>
</ol>

<h2>Обязательные ежедневные ритуалы</h2>
<ul>
<li><strong>Пить достаточно воды</strong> — 8 стаканов в день</li>
<li><em>Делать перерывы</em> при работе за компьютером</li>
<li>Практиковать <b>благодарность</b> — записывать 3 хорошие вещи за день</li>
<li><i>Отключаться</i> от гаджетов за час до сна</li>
</ul>

<hr>
<p><em>План составлен на: сегодняшняя дата</em></p>
<p><strong>Девиз дня:</strong> "Каждый маленький шаг приближает к большой цели!"</p>
</body>
</html>
```

Часть 6: Практическое задание для закрепления

Задание 1: Персонализируйте страницу увлечений

Создайте файл my_hobbies.html и заполните его СВОИМИ увлечениями:

```
html
<!DOCTYPE html>
<html lang="ru">
<head>
<title>Мои увлечения</title>
<meta charset="utf-8">
</head>
<body>
<h1>Мои уникальные увлечения</h1>

<h2>Что я люблю делать</h2>
<ul>
<li>[Ваше увлечение 1]</li>
<li>[Ваше увлечение 2]</li>
```

```
<li>[Ваше увлечение 3]</li>
<li>[Ваше увлечение 4]</li>
</ul>

<h2>Подробнее о моих интересах</h2>
<!-- Добавьте вложенные списки --&gt;

&lt;h2&gt;Чему я хочу научиться&lt;/h2&gt;
&lt;ul&gt;
&lt;li&gt;[Новый навык 1]&lt;/li&gt;
&lt;li&gt;[Новый навык 2]&lt;/li&gt;
&lt;/ul&gt;
&lt;/body&gt;
&lt;/html&gt;</pre>
```

Задание 2: Создайте свой распорядок дня

Создайте файл my_schedule.html:

```
html
<!DOCTYPE html>
<html lang="ru">
<head>
<title>Мой распорядок дня</title>
<meta charset="utf-8">
</head>
<body>
<h1>Мой идеальный день</h1>

<h2>Утро</h2>
<ol>
<li>[Первое утреннее дело]</li>
<li>[Второе утреннее дело]</li>
</ol>

<h2>День</h2>
<ol start="3">
<li>[Основная дневная задача]</li>
<li>[Вторая задача]</li>
</ol>

<h2>Вечер</h2>
<ol start="5">
<li>[Вечерние занятия]</li>
<li>[Подготовка ко сну]</li>
```

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

Часть 7: Проверка и тестирование

Что проверить после создания страниц:

Откройте оба файла в браузере

Убедитесь, что:

Все списки отображаются правильно

Нумерация в нумерованных списках работает корректно

Вложенные списки имеют правильные отступы

Текст читается легко

Все ссылки работают (если добавляли)

Протестируйте на разных устройствах (если возможно):

На компьютере

На планшете

На телефоне

Чек-лист качества:

Все теги правильно закрыты

Русские буквы отображаются без искажений

Списки имеют четкую структуру

Вложенные списки правильно оформлены

Текст хорошо читается

Страницы загружаются быстро

Часть 8: Дополнительные улучшения

Добавьте простые стили для красоты:

```
html
<head>
<title>Мои увлечения</title>
<meta charset="utf-8">
<style>
```

```
body {
  font-family: Arial, sans-serif;
  max-width: 800px;
```

```
margin: 0 auto;  
padding: 20px;  
line-height: 1.6;  
}  
  
h1 {  
color: #2c3e50;  
border-bottom: 2px solid #3498db;  
padding-bottom: 10px;  
}  
  
h2 {  
color: #34495e;  
}  
  
ul, ol {  
background-color: #f8f9fa;  
padding: 15px 15px 15px 40px;  
border-radius: 5px;  
}  
</style>  
</head>
```

Заключение

Что Вы достигли в этом уроке:

-  **Создали две практические страницы** с реальным содержанием
-  **Научились комбинировать** маркированные и нумерованные списки
-  **Освоили создание многоуровневых** вложенных списков
-  **Поняли разницу** между ситуациями для `` и ``
-  **Закрепили использование** тегов форматирования внутри списков

Ваши страницы теперь:

-  **Практически полезны** — Вы можете использовать их в повседневной жизни
-  **Хорошо структурированы** — информация подана логично и понятно
-  **Доступны** — правильная семантика помогает скринридерам
-  **Профессионально выглядят** — благодаря правильной разметке

Следующие шаги:

Покажите свои страницы друзьям или родным
Используйте страницу с распорядком дня на практике

Регулярно обновляйте список увлечений
Экспериментируйте с добавлением новых разделов

Теперь Вы не только знаете теорию списков, но и умеете применять их для решения реальных задач! Эти навыки пригодятся Вам как в веб-разработке, так и в повседневной организации информации.

Глава 7: Ссылки — "двери" в другие страницы

7.1. Волшебный тег <a> и его атрибут href

Введение: Создаём "двери" между веб-страницами

Тег <a> (от английского "anchor" — якорь) — это один из самых важных и волшебных тегов в HTML. Он превращает обычный текст в кликабельные ссылки, которые соединяют веб-страницы в единую "паутину". Без ссылок не было бы и Всемирной паутины!

Часть 1: Что такое тег <a> и атрибут href?

Базовая структура ссылки:

```
html  
<a href="адрес">текст ссылки</a>
```

Разберем по частям:

<a> — тег-контейнер для создания ссылки.
href (hypertext reference) — атрибут, указывающий **куда** ведет ссылка.
Текст ссылки — видимая часть, которую кликает пользователь.

Простой пример:

```
html  
<a href="https://yandex.ru">Перейти на Яндекс</a>
```

Результат: При клике на данную ссылку Вы попадёте на главную страницу портала Яндекс.

Часть 2: Виды ссылок — куда можно вести пользователя

1. Ссылка на внешний сайт

```
html  
<a href="https://www.google.com">Поиск в Google</a>  
<a href="https://ru.wikipedia.org">Википедия</a>  
<a href="https://www.youtube.com">Смотреть видео на YouTube</a>
```

2. Ссылка на другую страницу Вашего сайта

```
html
<a href="about.html">Обо мне</a>
<a href="hobbies.html">Мои увлечения</a>
<a href="contacts.html">Контакты</a>
```

3. Ссылка на email

```
html
<a href="mailto: example@mail.ru">Написать мне письмо</a>
```

При клике откроется почтовая программа

4. Ссылка на телефонный номер

```
html
<a href="tel: +71234567890">Позвонить мне</a>
```

На мобильных устройствах откроется набор номера

Часть 3: Атрибуты тега <a>

target — где открыть ссылку

```
html

<a href="page.html">Обычная ссылка</a>


<a href="https://example.com" target="_blank">Открыть в новой
вкладке</a>


<a href="page.html" target="_parent">В родительском окне</a>
```

Практический пример:

```
html
<a href="https://htmlbook.ru" target="_blank">Справочник HTML (откроется
в новой вкладке)</a>
```

title — всплывающая подсказка

```
html
<a href="garden.html" title="Посмотрите фото моего сада">Мой сад</a>
```

При наведении курсора покажется подсказка

Часть 4: Практические примеры использования

Пример 1: Навигационное меню сайта

```
html
<nav>
<h3>Навигация по сайту:</h3>
<ul>
<li><a href="index.html">Главная</a></li>
<li><a href="about.html">Обо мне</a></li>
<li><a href="hobbies.html">Мои увлечения</a></li>
<li><a href="contacts.html">Контакты</a></li>
</ul>
</nav>
```

Пример 2: Ссылки в тексте

```
html
<p>Я учу HTML по <a href="https://htmlbook.ru" target="_blank">онлайн-справочнику</a> и смотрю видеоуроки на <a href="https://www.youtube.com">YouTube</a>. Если у Вас есть вопросы, <a href="mailto: moyarochta@example.com">напишите мне</a>.</p>
```

Пример 3: Ссылки на полезные ресурсы

```
html
<h3>Полезные ресурсы для учебы:</h3>
<ul>
<li><a href="https://validator.w3.org" target="_blank" title="Проверка HTML-кода">W3C Validator</a> — проверка валидности кода</li>
<li><a href="https://www.w3schools.com" target="_blank">W3Schools</a> — уроки по веб-технологиям</li>
<li><a href="https://developer.mozilla.org" target="_blank">MDN Web Docs</a> — документация для разработчиков</li>
</ul>
```

Часть 5: Относительные и абсолютные пути

Абсолютные пути (полный адрес)

```
html
<a href="https://www.example.com/folder/page.html">Полная ссылка</a>
<a href="https://yandex.ru">Внешний сайт</a>
```

Относительные пути (относительно текущей страницы)

```
html
<!-- Файл в той же папке -->
Страница в этой папке

<!-- Файл в подпапке -->
Страница в папке pages

<!-- Файл на уровень выше -->
Страница в родительской папке

<!-- Файл в соседней папке -->
Страница в другой папке
```

Визуализация структуры папок:

```
text
Мой первый сайт/
    ├── index.html
    ├── about.html
    └── hobbies/
        ├── garden.html
        ├── books.html
        └── contacts.html
```

Примеры ссылок для этой структуры:

```
html
<!-- Из index.html -->
Обо мне
Мой сад
Контакты

<!-- Из hobbies/garden.html -->
На главную
Обо мне
Мои книги
```

Часть 6: Особые виды ссылок

Якорные ссылки (внутри страницы)

```
html
<!-- Создаем якорь -->
```

```
<h2 id="chapter1">Глава 1</h2>
<p>Текст первой главы...</p>

<h2 id="chapter2">Глава 2</h2>
<p>Текст второй главы...</p>

<!-- Ссылки на якоря -->
<a href="#chapter1">Перейти к главе 1</a>
<a href="#chapter2">Перейти к главе 2</a>
```

Ссылка для скачивания файла

```
html
<a href="document.pdf" download>Скачать PDF</a>
<a href="photo.jpg" download="my_photo.jpg">Скачать фотографию</a>
```

Ссылка-кнопка

```
html
<a href="order.html" style="display: inline-block; padding: 10px 20px;
background: blue; color: white; text-decoration: none;">Заказать сейчас</a>
```

Часть 7: Практическое задание

Создайте страницу с различными типами ссылок:

```
html
<!DOCTYPE html>
<html lang="ru">
<head>
<title>Практика с ссылками</title>
<meta charset="utf-8">
</head>
<body>
<h1>Мои полезные ссылки</h1>

<h2>Навигация по моему сайту</h2>
<ul>
<li><a href="index.html">Главная страница</a></li>
<li><a href="about.html">Информация обо мне</a></li>
<li><a href="hobbies/garden.html">Мой сад</a></li>
<li><a href="hobbies/books.html">Мои книги</a></li>
<li><a href="contacts.html">Как со мной связаться</a></li>
</ul>
```

```
<h2>Полезные образовательные ресурсы</h2>
<ul>
<li><a href="https://htmlbook.ru" target="_blank" title="Справочник по HTML и CSS">HTML Book</a> — мой главный помощник в учебе</li>
<li><a href="https://validator.w3.org" target="_blank">W3C Validator</a> — проверяю здесь свой код</li>
<li><a href="https://www.w3schools.com" target="_blank">W3Schools</a> — смотрю примеры кода</li>
<li><a href="https://ru.wikipedia.org" target="_blank">Википедия</a> — нахожу информацию для статей</li>
</ul>
```

```
<h2>Ссылки для связи</h2>
<ul>
<li><a href="mailto: ivanov@example.com">Написать мне на email</a></li>
<li><a href="tel: +79123456789">Позвонить по телефону</a></li>
</ul>
```

```
<h2>Быстрая навигация по странице</h2>
<p>
<a href="#resources">Образовательные ресурсы</a> |<br/>
<a href="#contacts">Контакты</a> |<br/>
<a href="#about">О проекте</a>
</p>
```

```
<!-- Якоря для навигации -->
<h3 id="resources">Образовательные ресурсы</h3>
<p>Здесь я собираю полезные сайты для обучения...</p>
```

```
<h3 id="contacts">Контакты</h3>
<p>Как со мной связаться...</p>

<h3 id="about">О проекте</h3>
<p>Информация о моем сайте...</p>
```

```
<p><a href="#">Наверх ↑</a></p>
</body>
</html>
```

Часть 8: Стилизация ссылок

CSS-свойства для оформления ссылок:

```
html
<head>
```

```
<style>
/* Базовая стилизация ссылок */
a {
color: #0066cc;
text-decoration: none;
}

/* При наведении */
a:hover {
color: #ff6600;
text-decoration: underline;
}

/* Посещенные ссылки */
a:visited {
color: #663399;
}

/* Активные ссылки (при нажатии) */
a:active {
color: #ff0000;
}

/* Стили для навигационных ссылок */
.nav-link {
display: inline-block;
padding: 10px 15px;
background: #f0f0f0;
border-radius: 5px;
margin: 5px;
}

.nav-link:hover {
background: #e0e0e0;
}
</style>
</head>
```

Часть 9: Частые ошибки и лучшие практики

✖ НЕПРАВИЛЬНО:

```
html
<!-- Забыли атрибут href -->
<a>Просто текст</a>
```

```
<!-- Неправильный путь -->
<a href="C:\Users\Name\file.html">Локальный путь</a>

<!-- Ссылка без осмысленного текста -->
<a href="page.html">кликните здесь</a>

<!-- Длинные нечитаемые URL как текст -->
<a href="page.html">https://www.mysite.com/folder/subfolder/page.html</a>
```

ПРАВИЛЬНО:

```
html
<!-- Правильная структура -->
<a href="about.html">Обо мне</a>

<!-- Осмысленный текст ссылки -->
<a href="gardening-tips.html">Полезные советы по садоводству</a>

<!-- Внешние ссылки с target="_blank" -->
<a href="https://example.com" target="_blank">Пример сайта</a>

<!-- С подсказкой -->
<a href="photo.jpg" title="Моя фотография сада">Посмотреть фото</a>
```

Лучшие практики:

Осмысленный текст — понятно, куда ведет ссылка
target="_blank" для внешних сайтов
title для дополнительной информации
Относительные пути для внутренних страниц
Проверка работоспособности всех ссылок

Часть 10: Проверка и тестирование ссылок

Чек-лист после создания ссылок:

Все ссылки ведут куда нужно
Внешние ссылки открываются в новой вкладке
Относительные пути работают правильно
Email и телефонные ссылки функционируют
Текст ссылок понятен и информативен
Нет "битых" ссылок (ошибка 404)

Тестирование на разных устройствах:

Компьютер (разные браузеры)
Планшет
Смартфон
Проверка на разных разрешениях экрана

Заключение

Что Вы теперь умеете:

- Создавать различные типы ссылок
- Правильно использовать атрибут href
- Работать с относительными и абсолютными путями
- Добавлять ссылки для email и телефонов
- Создавать навигацию между страницами
- Использовать дополнительные атрибуты (**target, title**)

Ключевые преимущества ссылок:

-  Соединяют страницы в единую сеть
-  Организуют навигацию по сайту
-  Обеспечивают связь с посетителями
-  Интегрируют с внешними ресурсами
- ❖ Улучшают доступность сайта

Запомните: Ссылки — это "кровеносная система" веба. Они превращают набор отдельных страниц в единую, взаимосвязанную структуру, которую мы называем Всемирной паутиной.

Теперь Ваш сайт обретает настоящую жизнь — страницы соединяются друг с другом, создавая удобную навигацию и полноценную веб-структуру!

7.2. Как сделать ссылку на другой сайт (например, на любимый новостной портал)

Введение: Выходим за пределы своего сайта

Ссылки на внешние сайты — это как двери в другие "дома" интернета. Они позволяют Вашим посетителям переходить на полезные ресурсы, делиться дополнительной информацией и расширять их опыт после посещения Вашего сайта.

Часть 1: Базовая структура внешней ссылки

Основной синтаксис:

html

```
<a href="https://адрес-сайта.py">Название сайта</a>
```

Ключевые особенности внешних ссылок:

Всегда начинаются с https:// или http://

Содержат **полный адрес** сайта

Рекомендуется открывать в **новой вкладке** (target="_blank")

Часть 2: Пошаговая инструкция создания внешней ссылки

Шаг 1: Находим адрес сайта

Откройте любимый новостной портал и скопируйте адрес из адресной строки браузера:

text

```
https://ria.ru      // РИА Новости  
https://www.vesti.ru // Вести.ru  
https://lenta.ru    // Лента.ru  
https://rg.ru       // Российская газета
```

Шаг 2: Создаем базовую ссылку

html

```
<a href="https://ria.ru">РИА Новости</a>
```

Шаг 3: Добавляем открытие в новой вкладке

html

```
<a href="https://ria.ru" target="_blank">РИА Новости</a>
```

Шаг 4: Добавляем всплывающую подсказку

html

```
<a href="https://ria.ru" target="_blank" title="Читайте последние новости на РИА Новости">РИА Новости</a>
```

Часть 3: Практические примеры популярных новостных порталов

Пример 1: Основные государственные СМИ

html

```
<h3>Государственные новостные порталы:</h3>
<ul>
<li><a href="https://ria.ru" target="_blank" title="Федеральное агентство новостей">РИА Новости</a></li>
<li><a href="https://tass.ru" target="_blank" title="Информационное агентство России">ТАСС</a></li>
<li><a href="https://rg.ru" target="_blank" title="Официальное издание правительства">Российская газета</a></li>
<li><a href="https://www.vesti.ru" target="_blank" title="Новости телеканала Россия">Вести.ru</a></li>
</ul>
```

Пример 2: Независимые новостные ресурсы

html

```
<h3>Независимые новостные сайты:</h3>
<ul>
<li><a href="https://lenta.ru" target="_blank" title="Ежедневная интернет-газета">Лента.ру</a></li>
<li><a href="https://www.gazeta.ru" target="_blank" title="Ежедневная общественно-политическая газета">Газета.Ru</a></li>
<li><a href="https://www.kommersant.ru" target="_blank" title="Деловая газета Коммерсантъ">Коммерсантъ</a></li>
<li><a href="https://www.rbc.ru" target="_blank" title="Деловые новости и аналитика">РБК</a></li>
</ul>
```

Пример 3: Региональные и тематические новости

html

```
<h3>Тематические новостные ресурсы:</h3>
<ul>
```

```
<li><a href="https://www.sport-express.ru" target="_blank" title="Спортивные  
новости">Спорт-Экспресс</a></li>  
<li><a href="https://www.kp.ru" target="_blank" title="Новости и  
события">Комсомольская правда</a></li>  
<li><a href="https://aif.ru" target="_blank" title="Общественно-политическая  
газета">Аргументы и факты</a></li>  
<li><a href="https://www.mk.ru" target="_blank" title="Ежедневная  
газета">Московский комсомолец</a></li>  
</ul>
```

Часть 4: Полная страница с новостными ссылками

Создайте файл news_links.html:

```
html  
<!DOCTYPE html>  
<html lang="ru">  
<head>  
<title>Мои любимые новостные порталы</title>  
<meta charset="utf-8">  
<style>  
body {  
font-family: Arial, sans-serif;  
max-width: 800px;  
margin: 0 auto;  
padding: 20px;  
line-height: 1.6;  
}  
h1 {  
color: #2c3e50;  
text-align: center;  
border-bottom: 2px solid #3498db;  
padding-bottom: 10px;  
}  
h2 {  
color: #34495e;  
background: #f8f9fa;  
padding: 10px;  
border-left: 4px solid #3498db;  
}  
ul {  
background: #f8f9fa;  
padding: 15px 15px 15px 40px;  
border-radius: 5px;  
}
```

```
a {
color: #2980b9;
text-decoration: none;
transition: color 0.3s;
}
a:hover {
color: #e74c3c;
text-decoration: underline;
}
.note {
background: #fff3cd;
padding: 15px;
border-left: 4px solid #ffc107;
margin: 20px 0;
}
</style>
</head>
<body>
<h1>💻 Моя коллекция новостных порталов</h1>

<div class="note">
<strong>Примечание:</strong> Все ссылки открываются в новой вкладке,
чтобы Вы не потеряли мою страницу!
</div>

<h2>Российские основные государственные СМИ</h2>
<ul>
<li>
<a href="https://ria.ru" target="_blank" title="Федеральное агентство
новостей">
<strong>РИА Новости</strong>
</a> — одно из крупнейших информационных агентств России
</li>
<li>
<a href="https://tass.ru" target="_blank" title="Информационное агентство
России">
<strong>ТАСС</strong>
</a> — ведущее государственное информационное агентство
</li>
<li>
<a href="https://rg.ru" target="_blank" title="Официальное издание
правительства">
<strong>Российская газета</strong>
</a> — официальный печатный орган правительства РФ
</li>
```

```
<li>
<a href="https://www.vesti.ru" target="_blank" title="Новости телеканала
Россия">
<strong>Вести.ru</strong>
</a> — новостной портал телеканала "Россия"
</li>
</ul>
```

```
<h2>▣ Независимые новостные ресурсы</h2>
<ul>
<li>
<a href="https://lenta.ru" target="_blank" title="Ежедневная интернет-
газета">
<strong>Лента.ру</strong>
</a> — одна из первых и самых популярных интернет-газет
</li>
<li>
<a href="https://www.gazeta.ru" target="_blank" title="Ежедневная
общественно-политическая газета">
<strong>Газета.Ru</strong>
</a> — ежедневная электронная газета
</li>
<li>
<a href="https://www.kommersant.ru" target="_blank" title="Деловая газета
Коммерсантъ">
<strong>Коммерсантъ</strong>
</a> — влиятельная деловая газета
</li>
<li>
<a href="https://www.rbc.ru" target="_blank" title="Деловые новости и
аналитика">
<strong>РБК</strong>
</a> — деловые новости и аналитические материалы
</li>
</ul>
```

```
<h2>🎯 Тематические и региональные новости</h2>
<ul>
<li>
<a href="https://www.sport-express.ru" target="_blank" title="Спортивные
новости">
<strong>Спорт-Экспресс</strong>
</a> — самые свежие спортивные новости
</li>
<li>
```

```
<a href="https://www.kp.ru" target="_blank" title="Новости и события">
<strong>Комсомольская правда</strong>
</a> — популярная ежедневная газета
</li>
<li>
<a href="https://aif.ru" target="_blank" title="Общественно-политическая
газета">
<strong>Аргументы и факты</strong>
</a> — общественно-политическая газета
</li>
<li>
<a href="https://www.mk.ru" target="_blank" title="Ежедневная газета">
<strong>Московский комсомолец</strong>
</a> — ежедневная газета с широкой тематикой
</li>
</ul>
```

```
<h2>🌐 Международные новости на русском</h2>
<ul>
<li>
...
</li>
</ul>
```

```
<h2>🔧 Полезные инструменты для веб-разработки</h2>
<ul>
<li>
<a href="https://validator.w3.org" target="_blank" title="Проверка HTML-
кода">
<strong>W3C Validator</strong>
</a> — проверяю здесь правильность своего HTML-кода
</li>
<li>
<a href="https://htmlbook.ru" target="_blank" title="Справочник по HTML и
CSS">
<strong>HTML Book</strong>
</a> — мой главный помощник в изучении HTML
</li>
</ul>
```

```
<div class="note">
<strong>Совет:</strong> Добавляйте сюда свои любимые сайты и
регулярно обновляйте список!
</div>
```

```
<hr>
<p><em>Страница создана: [сегодняшняя дата] | Последнее
обновление: [сегодняшняя дата]</em></p>
<p><a href="index.html">← Вернуться на главную страницу</a></p>
</body>
</html>
```

Часть 5: Особые случаи и нюансы

Ссылки на конкретные разделы сайтов:

```
html
<!-- На конкретную статью -->
<a href="https://ria.ru/20240520/novost-1954321654.html" target="_blank">
Читать новость на РИА Новости
</a>

<!-- На конкретную рубрику -->
<a href="https://www.vesti.ru/finance" target="_blank">
Финансовые новости на Вестях
</a>

<!-- На поиск по сайту -->
<a href="https://lenta.ru/search?query=HTML" target="_blank">
Поиск статей про HTML на Ленте.ру
</a>
```

Ссылки с UTM-метками (для аналитики):

```
html
<a href="https://ria.ru?utm_source=mysite&utm_medium=referral"
target="_blank">
РИА Новости (реферальная ссылка)
</a>
```

Часть 6: Практическое задание

Создайте свою персонализированную страницу с ссылками:

```
html
<!DOCTYPE html>
<html lang="ru">
<head>
<title>Мои любимые сайты</title>
<meta charset="utf-8">
```

```
</head>
<body>
<h1>⭐ Мои любимые сайты в интернете</h1>

<h2>📰 Новостные порталы, которые я читаю</h2>
<ul>
<li>
<a href="[адрес Вашего любимого новостного сайта 1]" target="_blank">
[Название сайта 1]
</a> — [краткое описание, почему нравится]
</li>
<li>
<a href="[адрес Вашего любимого новостного сайта 2]" target="_blank">
[Название сайта 2]
</a> — [краткое описание, почему нравится]
</li>
</ul>

<h2>🎓 Образовательные ресурсы</h2>
<ul>
<li>
<a href="[адрес образовательного сайта]" target="_blank">
[Название сайта]
</a> — [что изучаю/изучал]
</li>
</ul>

<h2>♫ Развлечения и хобби</h2>
<ul>
<li>
<a href="[адрес сайта по хобби]" target="_blank">
[Название сайта]
</a> — [чем занимаюсь/увлекаюсь]
</li>
</ul>

<h2>🛒 Полезные сервисы</h2>
<ul>
<li>
<a href="[адрес полезного сайта]" target="_blank">
[Название сайта]
</a> — [для чего использую]
</li>
</ul>
</body>
```

</html>

Часть 7: Лучшие практики и рекомендации

✓ ЧТО ДЕЛАТЬ:

html

<!-- Всегда используйте https:// -->
Правильно

<!-- Добавляйте target="_blank" для внешних ссылок -->
Откроется в новой
вкладке

<!-- Используйте осмысленные описания -->
РИА Новости - государственное
агентство новостей

<!-- Проверяйте ссылки на работоспособность -->
<!-- Перед публикацией убедитесь, что сайт доступен -->

✗ ЧЕГО ИЗБЕГАТЬ:

html

<!-- Не забывайте про https:// -->
Неправильно (без протокола)

<!-- Не используйте не информативные тексты -->
тут

<!-- Не ведите на небезопасные сайты -->
Небезопасно (http вместо https)

Проверка ссылок:

Тестируйте все ссылки после создания

Убедитесь, что сайты доступны

Проверьте, что открываются в новой вкладке

Убедитесь, что описания точны и полезны

Часть 8: Дополнительные возможности

Добавление иконок к ссылкам:

html

<style>

```
.external-link::after {  
content: "↗";  
font-size: 0.8em;  
}  
</style>
```

[РИА Новости](https://ria.ru)

Группировка ссылок по категориям:

html

```
<div class="news-category">  
<h3>Государственные СМИ</h3>  
<div class="links">  
<a href="https://ria.ru" target="_blank">РИА Новости</a>  
<a href="https://tass.ru" target="_blank">ТАСС</a>  
</div>  
</div>
```

Заключение

Что Вы теперь умеете:

- Создавать ссылки на внешние сайты
- Правильно формировать URL-адреса
- Использовать `target="_blank"` для внешних ссылок
- Добавлять информативные описания
- Создавать структурированные списки ссылок
- Стилизовать ссылки для лучшего восприятия

Преимущества внешних ссылок:

-  **Расширяют возможности** вашего сайта
-  **Предоставляют дополнительную информацию**
-  **Устанавливают связи** с авторитетными ресурсами
-  **Улучшают пользовательский опыт**
-  **Могут положительно влиять на SEO**

Запомните: Внешние ссылки — это проявление щедрости в интернете. Вы делитесь с посетителями полезными ресурсами и помогаете им находить дополнительную информацию beyond Вашего сайта.

Теперь Ваш сайт стал частью большой веб-экосистемы, связанной с полезными и авторитетными ресурсами интернета!

7.3. Как сделать ссылку на другую свою страницу

Введение: Создаём навигацию внутри своего сайта

Ссылки между Вашиими собственными страницами — это как комнаты в одном доме. Они создают логичную структуру сайта и позволяют посетителям легко перемещаться между разделами без использования кнопки "Назад" в браузере.

Часть 1: Базовые принципы внутренних ссылок

Основное отличие от внешних ссылок:

Не требуют `https://`

Используют **относительные пути**

Обычно открываются в **текущей вкладке**

Ведут на страницы **вашего собственного сайта**

Базовая структура:

html

```
<a href="имя-файла.html">Название страницы</a>
```

Часть 2: Структура папок проекта

Прежде чем создавать ссылки, давайте организуем наши файлы:

text

Мой первый сайт/

```
    └── index.html      (Главная страница)
    └── about.html     (Обо мне)
    └── hobbies/
        ├── garden.html   (Мой сад)
        ├── books.html     (Мои книги)
        └── cooking.html    (Моя кухня)
    └── contacts.html   (Контакты)
    └── resources/
        └── links.html     (Мои ссылки)
```

Часть 3: Типы относительных путей

1. Файл в той же папке (самый простой случай)

html

```
<!-- Из index.html ссылаемся на about.html -->
```

```
<a href="about.html">Обо мне</a>
<a href="contacts.html">Контакты</a>
```

2. Файл в подпапке

```
html
<!-- Из index.html ссылаемся на garden.html в папке hobbies -->
<a href="hobbies/garden.html">Мой сад</a>
<a href="hobbies/books.html">Мои книги</a>
<a href="hobbies/cooking.html">Моя кухня</a>
```

3. Файл в родительской папке

```
html
<!-- Из hobbies/garden.html ссылаемся на index.html -->
<a href="../index.html">На главную</a>
<a href="../about.html">Обо мне</a>
```

4. Файл в соседней папке

```
html
<!-- Из hobbies/garden.html ссылаемся на links.html в папке resources -->
<a href="../resources/links.html">Полезные ссылки</a>
```

Часть 4: Практические примеры навигации

Пример 1: Навигационное меню для главной страницы

В файле index.html:

```
html
<nav>
<h2>Навигация по сайту:</h2>
<ul>
<li><a href="index.html">Главная</a></li>
<li><a href="about.html">Обо мне</a></li>
<li><a href="hobbies/garden.html">Мой сад</a></li>
<li><a href="hobbies/books.html">Мои книги</a></li>
<li><a href="hobbies/cooking.html">Моя кухня</a></li>
<li><a href="resources/links.html">Полезные ссылки</a></li>
<li><a href="contacts.html">Контакты</a></li>
</ul>
</nav>
```

Пример 2: Навигация внутри раздела "Увлечения"

В файле hobbies/garden.html:

```
html
<nav>
<h3>Разделы увлечений:</h3>
<ul>
<li><a href="../index.html">Главная</a></li>
<li><a href="../about.html">Обо мне</a></li>
<li><strong>Мой сад</strong></li>
<li><a href="books.html">Мои книги</a></li>
<li><a href="cooking.html">Моя кухня</a></li>
<li><a href="../contacts.html">Контакты</a></li>
</ul>
</nav>
```

Часть 5: Полный пример сайта с навигацией

Файл index.html (Главная страница):

```
html
<!DOCTYPE html>
<html lang="ru">
<head>
<title>Мой личный сайт - Главная</title>
<meta charset="utf-8">
<style>
nav {
background: #f4f4f4;
padding: 15px;
border-radius: 5px;
margin-bottom: 20px;
}
nav ul {
list-style: none;
padding: 0;
margin: 0;
display: flex;
gap: 15px;
}
nav a {
text-decoration: none;
color: #333;
padding: 5px 10px;
border-radius: 3px;
}
nav a:hover {
```

```
background: #ddd;
}

.current {
font-weight: bold;
color: #0066cc;
}

</style>
</head>
<body>
<header>
<h1>✿ Добро пожаловать на мой сайт!</h1>

<nav>
<ul>
<li><a href="index.html" class="current">Главная</a></li>
<li><a href="about.html">Обо мне</a></li>
<li><a href="hobbies/garden.html">Мой сад</a></li>
<li><a href="hobbies/books.html">Мои книги</a></li>
<li><a href="hobbies/cooking.html">Моя кухня</a></li>
<li><a href="contacts.html">Контакты</a></li>
</ul>
</nav>
</header>

<main>
<h2>Рад Вас видеть на моём сайте!</h2>
<p>Это моя первая попытка создать собственный веб-сайт. Здесь я буду делиться:</p>

<ul>
<li>Своими <a href="hobbies/garden.html">увлечениями и хобби</a></li>
<li><a href="about.html">Историей</a> моего пути в веб-разработку</li>
<li>Полезными <a href="resources/links.html">ссылками и ресурсами</a></li>
</ul>

<h3>Новые добавления:</h3>
<ul>
<li>✿ Недавно добавил страницу <a href="hobbies/books.html">"Мои книги"</a></li>
<li>✿ Обновил раздел <a href="hobbies/cooking.html">"Моя кухня"</a></li>
<li>✿ Добавил <a href="contacts.html">контактную информацию</a></li>
</ul>
</main>
```

```
<footer>
<hr>
<p><small>© 2024 Мой первый сайт. Все права защищены.</small></p>
<p><a href="contacts.html">Связаться со мной</a> | <a
href="resources/links.html">Полезные ресурсы</a></p>
</footer>
</body>
</html>
```

Файл hobbies/garden.html (Страница о саде):

```
html
<!DOCTYPE html>
<html lang="ru">
<head>
<title>Мой сад - Мой личный сайт</title>
<meta charset="utf-8">
<style>
/* Тот же стиль, что и в index.html */
nav { background: #f4f4f4; padding: 15px; border-radius: 5px; margin-bottom: 20px; }
nav ul { list-style: none; padding: 0; margin: 0; display: flex; gap: 15px; }
nav a { text-decoration: none; color: #333; padding: 5px 10px; border-radius: 3px; }
nav a:hover { background: #ddd; }
.current { font-weight: bold; color: #0066cc; }
</style>
</head>
<body>
<header>
<h1> Мой любимый сад</h1>

<nav>
<ul>
<li><a href="../index.html">Главная</a></li>
<li><a href="../about.html">Обо мне</a></li>
<li><a href="garden.html" class="current">Мой сад</a></li>
<li><a href="books.html">Мои книги</a></li>
<li><a href="cooking.html">Моя кухня</a></li>
<li><a href="../contacts.html">Контакты</a></li>
</ul>
</nav>
</header>
```

```
<main>
<h2>Что растёт в моём саду</h2>

<h3>Цветы:</h3>
<ul>
<li>Розы (15 сортов)</li>
<li>Пионы</li>
<li>Тюльпаны</li>
<li>Георгины</li>
</ul>

<h3>Овощи:</h3>
<ul>
<li>Помидоры</li>
<li>Огурцы</li>
<li>Морковь</li>
<li>Картофель</li>
</ul>

<p>Также у меня есть страница про <a href="books.html">мои книги</a>
и <a href="cooking.html">кулинарные рецепты</a>.</p>
</main>
```

```
<footer>
<hr>
<p><small>© 2024 Мой первый сайт. <a href="../index.html">Вернуться на
главную</a></small></p>
</footer>
</body>
</html>
```

Часть 6: Создание недостающих страниц

Базовый шаблон для новых страниц:

```
html
<!DOCTYPE html>
<html lang="ru">
<head>
<title>Название страницы - Мой личный сайт</title>
<meta charset="utf-8">
<style>
nav { background: #f4f4f4; padding: 15px; border-radius: 5px; margin-bottom: 20px; }
nav ul { list-style: none; padding: 0; margin: 0; display: flex; gap: 15px; }
```

```
nav a { text-decoration: none; color: #333; padding: 5px 10px; border-radius: 3px; }
nav a:hover { background: #ddd; }
.current { font-weight: bold; color: #0066cc; }
</style>
</head>
<body>
<header>
<h1>Заголовок страницы</h1>

<nav>
<ul>
<li><a href="../index.html">Главная</a></li>
<li><a href="../about.html">Обо мне</a></li>
<li><a href="garden.html">Мой сад</a></li>
<li><a href="books.html">Мои книги</a></li>
<li><a href="cooking.html">Моя кухня</a></li>
<li><a href="../contacts.html">Контакты</a></li>
</ul>
</nav>
</header>

<main>
<!-- Содержимое страницы --&gt;
&lt;p&gt;Содержимое этой страницы...&lt;/p&gt;
&lt;/main&gt;

&lt;footer&gt;
&lt;hr&gt;
&lt;p&gt;&lt;small&gt;© 2024 Мой первый сайт. &lt;a href="../index.html"&gt;Вернуться на главную&lt;/a&gt;&lt;/small&gt;&lt;/p&gt;
&lt;/footer&gt;
&lt;/body&gt;
&lt;/html&gt;</pre>
```

Часть 7: Практическое задание

Задание: Создайте полноценный сайт с 5 страницами

Создайте структуру папок как показано выше

Создайте все файлы используя шаблоны

Наполните страницы контентом:

about.html — информация о вас
hobbies/books.html — Ваши любимые книги
hobbies/cooking.html — Ваши рецепты
contacts.html — как с Вами связаться

Пример содержимого для about.html:

```
html
<main>
<h2>Немного обо мне</h2>
<p>Меня зовут [Ваше имя], и я начинающий веб-разработчик.</p>

<h3>Мой путь в IT</h3>
<p>Всё началось с [ваша история]...</p>

<h3>Мои цели</h3>
<ul>
<li>Освоить HTML и CSS</li>
<li>Создать свой блог о <a href="hobbies/garden.html">садоводстве</a></li>
<li>Помогать другим осваивать технологии</li>
</ul>

<p>Также Вы можете посмотреть мои <a href="hobbies/books.html">любимые книги</a>
или <a href="contacts.html">связаться со мной</a>.</p>
</main>
```

Часть 8: Проверка и отладка ссылок

Чек-лист проверки ссылок:

Проверьте все ссылки в навигации:

Главная → Обо мне
Главная → Мой сад
Главная → Мои книги

Проверьте обратные ссылки:

Мой сад → Главная
Мои книги → Главная
Обо мне → Главная

Проверьте перекрёстные ссылки:

Обо мне → Мой сад ✓
Мой сад → Мои книги ✓
Мои книги → Контакты ✓

Частые ошибки и их решение:

✗ Ошибка: "Страница не найдена" (404)

```
html
<!-- Неправильно -->
<a href="hobies/garden.html">Мой сад</a> <!-- Опечатка в имени папки --
>

<!-- Правильно -->
<a href="hobbies/garden.html">Мой сад</a>
```

✗ Ошибка: Неправильный путь

```
html
<!-- Из hobbies/garden.html -->
<a href="index.html">Главная</a> <!-- Будет искать в папке hobbies -->

<!-- Правильно -->
<a href="../index.html">Главная</a>
```

✓ Правильная практика:

Всегда проверяйте пути
Используйте одинаковую структуру навигации
Тестируйте все ссылки после создания

Часть 9: Улучшение навигации

Добавление хлебных крошек:

```
html
<nav aria-label="Хлебные крошки">
<p>
<a href="../index.html">Главная</a> >
<a href="index.html">Увлечения</a> >
<span>Мой сад</span>
</p>
</nav>
```

Добавление кнопок "Вперёд/Назад":

html

```
<div class="navigation">
<a href="books.html">← Мои книги</a>
<a href="../index.html">Главная</a>
<a href="cooking.html">Моя кухня →</a>
</div>
```

Заключение

Что Вы теперь умеете:

- Создавать ссылки между своими страницами
- Правильно использовать относительные пути
- Строить многоуровневую структуру папок
- Создавать навигационные меню
- Организовывать логичную структуру сайта

Ключевые преимущества внутренних ссылок:

-  **Создают единое пространство** — посетители не теряются
-  **Улучшают навигацию** — легко перемещаться между разделами
-  **Увеличивают время на сайте** — посетители просматривают больше страниц
-  **Улучшают SEO** — поисковые системы лучше понимают структуру сайта
-  **Повышают доступность** — удобно для пользователей скринридеров

Золотые правила внутренних ссылок:

Единообразие — используйте одинаковую навигацию на всех страницах

Логичность — ссылки должны вести туда, куда ожидает пользователь

Проверка — всегда тестируйте все ссылки после создания

Доступность — делайте тексты ссылок понятными и информативными
Теперь Ваш сайт превратился из набора отдельных страниц в единую, взаимосвязанную систему с удобной навигацией и логичной структурой!

7.4. Практика: создаём вторую страницу "Мои любимые сайты" и связываем её с главной ссылками

Введение: Строим мосты между страницами

Сейчас мы создадим полноценный мини-сайт из двух страниц, связанных между собой ссылками. Это поможет закрепить все полученные знания о ссылках и навигации.

Часть 1: Подготовка структуры проекта

Создаем необходимые файлы:

Откройте папку "Мой первый сайт"

Создайте два файла:

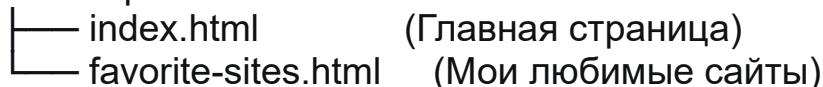
index.html (главная страница)

favorite-sites.html (страница с любимыми сайтами)

Структура нашего мини-сайта:

text

Мой первый сайт/



Часть 2: Создаём главную страницу (index.html)

html

```
<!DOCTYPE html>
<html lang="ru">
<head>
<title>Мой личный сайт - Главная</title>
<meta charset="utf-8">
<style>
body {
```

```
font-family: Arial, sans-serif;
max-width: 800px;
margin: 0 auto;
padding: 20px;
line-height: 1.6;
background-color: #f9f9f9;
}
```

```
header {
background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);
color: white;
```

```
padding: 30px;
border-radius: 10px;
text-align: center;
margin-bottom: 30px;
}
h1 {
margin: 0;
font-size: 2.5em;
}
nav {
background: white;
padding: 20px;
border-radius: 10px;
box-shadow: 0 2px 10px rgba(0,0,0,0.1);
margin-bottom: 30px;
}
.nav-links {
list-style: none;
padding: 0;
margin: 0;
display: flex;
justify-content: center;
gap: 20px;
}
.nav-links a {
text-decoration: none;
color: #333;
padding: 10px 20px;
border: 2px solid #667eea;
border-radius: 25px;
transition: all 0.3s ease;
}
.nav-links a:hover {
background: #667eea;
color: white;
}
.main-content {
background: white;
padding: 30px;
border-radius: 10px;
box-shadow: 0 2px 10px rgba(0,0,0,0.1);
}
.feature-card {
background: #f8f9fa;
padding: 20px;
```

```
border-radius: 8px;  
margin: 20px 0;  
border-left: 4px solid #667eea;  
}  
  
footer {  
text-align: center;  
margin-top: 40px;  
padding: 20px;  
color: #666;  
border-top: 1px solid #ddd;  
}  
  
.highlight {  
background: #fff3cd;  
padding: 15px;  
border-radius: 5px;  
border-left: 4px solid #ffc107;  
}  
  
</style>  
</head>  
<body>  
<header>  
<h1> Добро пожаловать на мой сайт!</h1>  
<p>Здесь я делясь своим путешествием в мир веб-разработки</p>  
</header>  
  
<nav>  
<ul class="nav-links">  
<li><a href="index.html"> Главная</a></li>  
<li><a href="favorite-sites.html"> Любимые сайты</a></li>  
</ul>  
</nav>  
  
<main class="main-content">  
<section>  
<h2>Обо мне</h2>  
<p>Привет! Меня зовут <strong>Александр</strong>, и я начинающий  
веб-разработчик. После выхода на пенсию я решил освоить  
современные технологии и научиться создавать сайты.</p>  
  
<div class="feature-card">  
<h3> Моя цель</h3>  
<p>Научиться создавать красивые и полезные веб-сайты, которые  
помогут другим людям освоить компьютерные технологии в зрелом  
возрасте.</p>  
</div>
```

```
</section>
```

```
<section>
<h2>Что я уже изучил</h2>
<ul>
<li>✓ Основы HTML и создание структуры страниц</li>
<li>✓ Работа с текстом: заголовки, абзацы, списки</li>
<li>✓ Создание ссылок и навигации между страницами</li>
<li>✓ Форматирование текста с помощью тегов</li>
</ul>
</section>
```

```
<section>
<h2>Мои проекты</h2>
<div class="feature-card">
<h3>💻 Страница с любимыми сайтами</h3>
<p>Я создал специальную страницу, где собрал все полезные ресурсы, которые помогают мне в обучении. Там Вы найдёте:</p>
<ul>
<li>Обучающие платформы по программированию</li>
<li>Справочники по HTML и CSS</li>
<li>Полезные инструменты для разработки</li>
<li>Новостные порталы о технологиях</li>
</ul>
<p><a href="favorite-sites.html" style="color: #667eea; font-weight: bold;">
 Посмотреть мою коллекцию сайтов →
</a></p>
</div>
</section>
```

```
<div class="highlight">
<h3>💡 Совет начинающим</h3>
<p>Не бойтесь начинать учиться в любом возрасте! Каждый день я узнаю что-то новое и убеждаюсь, что веб-разработка — это увлекательно и доступно для всех.</p>
</div>
</main>
```

```
<footer>
<p>📞 <strong>Свяжитесь со мной:</strong>
<a href="mailto:alexander@example.com">alexander@example.com</a></p>
<p>© 2024 Мой первый сайт. Создано с ❤ и энтузиазмом.</p>
<p><a href="favorite-sites.html">Перейти к моим любимым сайтам</a></p>
</footer>
```

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

Часть 3: Создаём страницу с любимыми сайтами (favorite-sites.html)

```
html
<!DOCTYPE html>
<html lang="ru">
<head>
<title>Мои любимые сайты - Мой личный сайт</title>
<meta charset="utf-8">
<style>
body {
font-family: Arial, sans-serif;
max-width: 900px;
margin: 0 auto;
padding: 20px;
line-height: 1.6;
background-color: #f9f9f9;
}
header {
background: linear-gradient(135deg, #ff6b6b 0%, #ee5a24 100%);
color: white;
padding: 30px;
border-radius: 10px;
text-align: center;
margin-bottom: 30px;
}
h1 {
margin: 0;
font-size: 2.5em;
}
nav {
background: white;
padding: 20px;
border-radius: 10px;
box-shadow: 0 2px 10px rgba(0,0,0,0.1);
margin-bottom: 30px;
}
.nav-links {
list-style: none;
padding: 0;
margin: 0;
display: flex;
justify-content: center;
```

```
gap: 20px;
}

.nav-links a {
text-decoration: none;
color: #333;
padding: 10px 20px;
border: 2px solid #ff6b6b;
border-radius: 25px;
transition: all 0.3s ease;
}

.nav-links a:hover {
background: #ff6b6b;
color: white;
}

.category {
background: white;
padding: 25px;
border-radius: 10px;
box-shadow: 0 2px 10px rgba(0,0,0,0.1);
margin-bottom: 30px;
}

.site-card {
background: #f8f9fa;
padding: 20px;
margin: 15px 0;
border-radius: 8px;
border-left: 4px solid #ff6b6b;
transition: transform 0.2s ease;
}

.site-card:hover {
transform: translateX(5px);
}

.external-link {
color: #ee5a24;
text-decoration: none;
font-weight: bold;
}

.external-link:hover {
text-decoration: underline;
}

.description {
color: #666;
margin: 5px 0 0 0;
font-size: 0.9em;
}
```

```
footer {
text-align: center;
margin-top: 40px;
padding: 20px;
color: #666;
border-top: 1px solid #ddd;
}
.back-link {
display: inline-block;
background: #ff6b6b;
color: white;
padding: 10px 20px;
border-radius: 25px;
text-decoration: none;
margin-top: 20px;
transition: background 0.3s ease;
}
.back-link:hover {
background: #ee5a24;
}
</style>
</head>
<body>
<header>
<h1>⭐ Мои любимые сайты</h1>
<p>Коллекция полезных ресурсов для обучения и развития</p>
</header>

<nav>
<ul class="nav-links">
<li><a href="index.html">🏠 Главная</a></li>
<li><a href="favorite-sites.html">⭐ Любимые сайты</a></li>
</ul>
</nav>

<main>
<section class="category">
<h2>🎓 Образовательные платформы</h2>

<div class="site-card">
<a href="https://htmlbook.ru" target="_blank" class="external-link">
 HTML Book
</a>
<p class="description">Лучший русскоязычный справочник по HTML и CSS. Пользуюсь им ежедневно для проверки тегов и свойств.</p>
```

```
</div>

<div class="site-card">
<a href="https://www.w3schools.com" target="_blank" class="external-link">
 W3Schools
</a>
<p class="description">Международная платформа с интерактивными
примерами кода. Отлично подходит для практики.</p>
</div>

<div class="site-card">
<a href="https://stepik.org" target="_blank" class="external-link">
 Stepik
</a>
<p class="description">Русскоязычная платформа с бесплатными курсами
по программированию. Прошёл здесь несколько курсов.</p>
</div>
</section>

<section class="category">
<h2> Инструменты для разработки</h2>

<div class="site-card">
<a href="https://validator.w3.org" target="_blank" class="external-link">
 W3C Validator
</a>
<p class="description">Проверяю здесь свой HTML-код на ошибки.
Помогает создавать качественные и валидные страницы.</p>
</div>

<div class="site-card">
<a href="https://codepen.io" target="_blank" class="external-link">
 CodePen
</a>
<p class="description">Онлайн-редактор для экспериментов с HTML, CSS
и JavaScript. Использую для быстрого тестирования идей.</p>
</div>

<div class="site-card">
<a href="https://github.com" target="_blank" class="external-link">
 GitHub
</a>
<p class="description">Храню здесь свои проекты и изучаю код других
разработчиков. Отличный способ учиться на реальных примерах.</p>
</div>
```

```
</section>

<section class="category">
<h2> Новости и сообщества</h2>

<div class="site-card">
<a href="https://habr.com" target="_blank" class="external-link">
 Habr
</a>
<p class="description">Крупнейшее IT-сообщество со статьями, новостями и обсуждениями. Читаю ежедневно для поддержания актуальных знаний.</p>
</div>

<div class="site-card">
<a href="https://tproger.ru" target="_blank" class="external-link">
 Tproger
</a>
<p class="description">Популярный портал о программировании с новостями, статьями и руководствами для начинающих.</p>
</div>

<div class="site-card">
<a href="https://stackoverflow.com" target="_blank" class="external-link">
 Stack Overflow
</a>
<p class="description">Когда сталкиваюсь с проблемой в коде, ищу решение здесь. Мировое сообщество разработчиков всегда готово помочь.</p>
</div>
</section>

<section class="category">
<h2> Дизайн и вдохновение</h2>

<div class="site-card">
<a href="https://dribbble.com" target="_blank" class="external-link">
 Dribbble
</a>
<p class="description">Смотрю работы дизайнеров для вдохновения. Помогает развивать чувство прекрасного в веб-дизайне.</p>
</div>

<div class="site-card">
<a href="https://awwwards.com" target="_blank" class="external-link">
```

```
<img alt="Awwwards trophy icon" data-bbox="98 68 125 91"/> Awwwards
</a>
<p class="description">Лучшие сайты мира. Изучаю тренды и
современные подходы к веб-разработке.</p>
</div>
</section>

<div style="text-align: center;">
<a href="index.html" class="back-link">
← Вернуться на главную страницу
</a>
</div>
</main>

<footer>
<p>⌚ <strong>Всего сайтов в коллекции:</strong> 10</p>
<p>© 2024 Моя коллекция полезных сайтов. Обновляется регулярно.</p>
<p><a href="index.html" style="color: #ff6b6b;">Вернуться на
главную</a></p>
</footer>
</body>
</html>
```

Часть 4: Анализ созданных ссылок

Ссылки на главной странице (index.html):

Навигационное меню:

```
html
<a href="index.html">🏠 Главная</a>
<a href="favorite-sites.html">⭐ Любимые сайты</a>
```

Внутри контента:

```
html
<a href="favorite-sites.html">🔗 Посмотреть мою коллекцию сайтов →</a>
```

В подвале:

```
html
<a href="favorite-sites.html">Перейти к моим любимым сайтам</a>
```

Ссылки на странице любимых сайтов (favorite-sites.html):

Навигационное меню:

html

```
<a href="index.html">🏠 Главная</a>
<a href="favorite-sites.html">⭐ Любимые сайты</a>
```

Внешние ссылки (открываются в новой вкладке):

html

```
<a href="https://htmlbook.ru" target="_blank" class="external-link">📚 HTML
Book</a>
```

Обратная ссылка:

html

```
<a href="index.html" class="back-link">← Вернуться на главную
страницу</a>
```

Часть 5: Практическое задание для закрепления

Задание 1: Создайте третью страницу

Добавьте страницу about.html с информацией о себе и свяжите её с главной:

html

```
<!-- В index.html добавьте в навигацию: -->
<li><a href="about.html">👤 Обо мне</a></li>
```

<!-- В about.html создайте обратные ссылки: -->

```
<a href="index.html">Главная</a>
<a href="favorite-sites.html">Любимые сайты</a>
```

Задание 2: Персонализируйте коллекцию сайтов

Замените примеры сайтов на СВОИ любимые ресурсы:

html

```
<div class="site-card">
<a href="[ВАШ_САЙТ]" target="_blank" class="external-link">
[НАЗВАНИЕ_САЙТА]
</a>
<p class="description">[ПОЧЕМУ Вам НРАВИТСЯ ЭТОТ САЙТ]</p>
</div>
```

Часть 6: Проверка работоспособности

Чек-лист тестирования:

Откройте index.html в браузере

Проверьте все ссылки:

Кликните "⭐ Любимые сайты" → должна открыться favorite-sites.html
На favorite-sites.html кликните "🏠 Главная" → вернуться на главную
Проверьте внешние ссылки (должны открываться в новых вкладках)

Убедитесь, что:

Все внутренние ссылки работают
Внешние ссылки открываются в новых вкладках
Навигация интуитивно понятна
Страницы загружаются быстро

Возможные проблемы и решения:

Проблема: "Страница не найдена" (404)

Решение: Проверьте правильность имени файла и путь

Проблема: Ссылка не работает

Решение: Убедитесь, что файл существует в указанном месте

Проблема: Внешние ссылки открываются в той же вкладке

Решение: Добавьте target="_blank" к внешним ссылкам

Часть 7: Дополнительные улучшения

Добавление счетчика посещений:

```
html
<div style="text-align: center; margin: 20px 0;">
<small>@@ Эту страницу посетили: <strong>1,247</strong> раз</small>
</div>
```

Добавление поиска по сайту:

```
html
<div style="text-align: center; margin: 20px 0;">
```

```
<input type="text" placeholder="Поиск по сайту..." style="padding: 10px; width: 300px;">
<button style="padding: 10px 20px;">Найти</button>
</div>
```

Заключение

Что Вы достигли в этом уроке:

- Создали полноценный двухстраничный сайт с взаимными ссылками
- Научились создавать навигационные меню между страницами
- Освоили разницу между внутренними и внешними ссылками
- Создали структурированную коллекцию полезных ресурсов
- Научились оформлять ссылки для лучшего пользовательского опыта

Ключевые навыки, которые Вы применили:

-  **Веб-дизайн** — создание привлекательного интерфейса
-  **Навигация** — логичная структура ссылок
-  **Семантика** — правильное использование HTML-тегов
-  **UX/UI** — удобство для пользователя
-  **Отладка** — проверка работоспособности ссылок

Ваш сайт теперь:

Функциональный — все ссылки работают корректно

Профессиональный — имеет единый стиль и логичную структуру

Полезный — содержит ценную коллекцию ресурсов

Масштабируемый — легко добавить новые страницы

Следующие шаги:

Покажите сайт друзьям и получите обратную связь

Добавьте новые страницы (о проектах, блог, контакты)

Регулярно обновляйте коллекцию любимых сайтов

Экспериментируйте с дизайном и оформлением

Теперь у Вас есть полноценный работающий веб-сайт, который можно развивать и улучшать! Это отличная основа для Вашего портфолио веб-разработчика.

Глава 8: Картинки — чтобы страница "заяграла"

8.1. Тег : атрибуты src (адрес картинки) и alt (описание для тех, кто не видит)

Введение: Оживляем сайт с помощью изображений

Тег — это один из самых волшебных тегов в HTML. Он позволяет добавлять на Ваши страницы фотографии, иллюстрации, диаграммы — всё, что делает контент наглядным и привлекательным. Давайте разберёмся, как правильно использовать этот тег.

Часть 1: Что такое тег и как он работает?

Базовая структура:

```
html  

```

Особенности тега :

Это **одиночный тег** (не требует закрывающего
Всегда требует **два основных атрибута**: **src** и **alt**
Загружает и отображает изображение на странице

Часть 2: Атрибут src (source — источник)

Что такое src:

Атрибут **src** указывает **путь к файлу изображения** — где браузер может найти картинку.

Способы указания пути:

1. Изображение в той же папке:

```
html  

```

2. Изображение в подпапке:

```
html  
  

```

3. Изображение из интернета (по URL):

```
html

```

4. Изображение на уровень выше:

```
html

```

Часть 3: Атрибут alt (alternative text — альтернативный текст)

Что такое alt и зачем он нужен:

Атрибут alt содержит **текстовое описание изображения**, которое:

Показывается если картинка не загрузилась

Озвучивается скринридерами для слабовидящих

Индексируется поисковыми системами

Помогает понять содержание при медленном интернете

Примеры хороших описаний:

✓ ПРАВИЛЬНО:

```
html



```

✗ НЕПРАВИЛЬНО:

```
html

```

Часть 4: Практические примеры использования

Пример 1: Фотография в статье

```
html
<article>
<h2>Мой сад в этом году</h2>
```

```
  
<p>В этом году мой сад особенно красив. Розы зацвели раньше  
обычного...</p>  
</article>
```

Пример 2: Фотография продукта

```
html  
<div class="product">  
  
<h3>Помидоры 'Бычье сердце'</h3>  
<p>Сочные и мясистые помидоры с моего огорода.</p>  
</div>
```

Пример 3: Логотип сайта

```
html  
<header>  
  
<h1>Сад и огород</h1>  
</header>
```

Пример 4: Декоративное изображение

```
html  
<div class="decoration">  
  
<!-- Для чисто декоративных изображений можно использовать<br/>пустой alt -->  
</div>
```

Часть 5: Подготовка изображений для сайта

Создаем структуру папок:

```
text  
Мой первый сайт/  
├── index.html  
├── images/      # Папка для изображений  
│   ├── photos/    # Фотографии  
│   │   ├── garden.jpg  
│   │   └── family.jpg
```

```
└── pets.jpg  
└── icons/      # Иконки  
    ├── phone.png  
    └── email.png  
└── backgrounds/ # Фоновые изображения  
    └── texture.jpg  
└── about.html
```

Рекомендации по форматам:

JPEG — для фотографий

PNG — для изображений с прозрачностью

GIF — для простой анимации

SVG — для иконок и логотипов

Часть 6: Дополнительные атрибуты тега

width и **height** — размеры изображения:

html

```
  

```

title — всплывающая подсказка:

html

```

```

loading — управление загрузкой:

html

```
  
<!-- lazy - отложенная загрузка, когда изображение появляется в зоне<br/>видимости -->
```

Часть 7: Практическое задание

Создайте страницу с галереей изображений:

html

```
<!DOCTYPE html>  
<html lang="ru">  
<head>  
<title>Моя фото-галерея</title>  
<meta charset="utf-8">  
<style>
```

```
body {
font-family: Arial, sans-serif;
max-width: 1000px;
margin: 0 auto;
padding: 20px;
background-color: #f5f5f5;
}

.gallery {
display: grid;
grid-template-columns: repeat(auto-fit, minmax(300px, 1fr));
gap: 20px;
margin-top: 30px;
}

.photo-card {
background: white;
padding: 15px;
border-radius: 10px;
box-shadow: 0 2px 10px rgba(0,0,0,0.1);
text-align: center;
}

.photo-card img {
max-width: 100%;
height: auto;
border-radius: 5px;
}

.photo-description {
margin-top: 10px;
color: #666;
font-size: 0.9em;
}

</style>
</head>
<body>
<h1>📸 Моя фото-галерея</h1>
<p>Здесь я собираю свои лучшие фотографии из разных сфер жизни.</p>

<div class="gallery">
<!-- Фотография 1 --&gt;
&lt;div class="photo-card"&gt;
&lt;img src="images/garden-spring.jpg"
alt="Весенний сад с цветущими тюльпанами и нарциссами"
width="400" height="300"&gt;
&lt;p class="photo-description"&gt;Весна в моём саду. Тюльпаны только начинают цвести.&lt;/p&gt;</pre>
```

```
</div>

<!-- Фотография 2 -->
<div class="photo-card">

<p class="photo-description">Мой фирменный яблочный пирог по
бабушкиному рецепту.</p>
</div>

<!-- Фотография 3 -->
<div class="photo-card">

<p class="photo-description">Моя домашняя библиотека. Люблю
перечитывать классику.</p>
</div>

<!-- Фотография 4 -->
<div class="photo-card">

<p class="photo-description">Наш кот Вася обожает спать на
солнышке.</p>
</div>
</div>

<div style="margin-top: 40px; text-align: center;">
<h2>Советы по созданию хороших alt-текстов</h2>
<ul style="text-align: left; display: inline-block;">
<li>✓ Описывайте что на изображении, а не как оно выглядит</li>
<li>✓ Будьте конкретны, но кратки</li>
<li>✓ Учитывайте контекст страницы</li>
<li>✓ Для декоративных изображений используйте alt=""</li>
</ul>
</div>
</body>
</html>
```

Часть 8: Частые ошибки и лучшие практики

✖ ЧАСТЫЕ ОШИБКИ:

Ошибка 1: Отсутствует атрибут alt

html

```
 <!-- НЕТ! Всегда добавляйте alt -->
```

Ошибка 2: Слишком общее описание

html

```
 <!-- Слишком общее -->
```

Ошибка 3: Использование имени файла

html

```
 <!-- Бесполезно -->
```

Ошибка 4: Избыточное описание

html

```

```

```
<!-- Слишком длинно -->
```

✓ ЛУЧШИЕ ПРАКТИКИ:

Практика 1: Конкретное и информативное описание

html

```

```

Практика 2: Учет контекста

html

```
<!-- В статье о садоводстве: -->
```

```

```

```
<!-- В статье о покупке инструментов: -->
```

```

```

Практика 3: Для декоративных изображений

```
html

<!-- Пустой alt для чисто декоративных элементов --&gt;</pre>
```

Часть 9: Специальные случаи

Изображения-ссылки:

```
html
<a href="large-photo.jpg">

</a>
```

Логотипы:

```
html
<a href="index.html">

</a>
```

Инфографика и диаграммы:

```
html

```

Скриншоты интерфейсов:

```
html

```

Часть 10: Проверка и тестирование

Чек-лист для изображений:

Все теги `` имеют атрибут `alt`
Описания `alt` информативны и релевантны
Пути `src` указаны правильно
Изображения загружаются корректно
Размеры изображений подходят для веба
Декоративные изображения имеют `alt=""`

Тестирование доступности:

Отключите изображения в браузере — должны показываться `alt`-тексты

Используйте скринридер — должны озвучиваться описания

Проверьте на медленном соединении — alt-тексты помогают понять содержание

Заключение

Что Вы теперь умеете:

- Добавлять изображения на веб-страницы
- Правильно указывать пути к файлам
- Создавать информативные alt-описания
- Различать контентные и декоративные изображения
- Оптимизировать изображения для веба

Ключевые преимущества правильного использования:

- ↳ **Доступность** — помочь слабовидящим пользователям
- 🔍 **SEO** — улучшение позиций в поисковых системах
- 📱 **Надежность** — резервный вариант при проблемах с загрузкой
- 💡 **Понятность** — ясное содержание для всех пользователей
- ⚡ **Производительность** — правильные размеры и форматы

Золотые правила работы с изображениями:

Всегда используйте атрибут alt

Будьте конкретны в описаниях

Оптимизируйте размеры файлов

Организуйте файлы в папках

Тестируйте на разных устройствах

Теперь Ваши веб-страницы станут по-настоящему живыми и наглядными! Изображения помогут лучше передавать Ваши мысли, делиться опытом и создавать привлекательный контент для посетителей.

8.2. Где найти простые и бесплатные картинки для тренировки

Введение: Безопасный и легальный поиск изображений

При создании сайтов очень важно использовать изображения правильно — не нарушая авторские права. К счастью, существует множество источников, где можно бесплатно и легально брать картинки для обучения и практики.

Часть 1: Бесплатные фотостоки с открытой лицензией

1. Unsplash — самый популярный и качественный

Сайт: unsplash.com

Особенности: Высококачественные фотографии профессионального уровня

Лицензия: Бесплатное использование, даже в коммерческих целях

Язык: Английский, но поиск по ключевым словам на русском работает

Как использовать:

html

```
<!-- Пример использования фото с Unsplash -->

```

Поисковые запросы для тренировки:

garden (сад)

flowers (цветы)

nature (природа)

books (книги)

cooking (готовка)

elderly people (пожилые люди)

2. Pexels — отличная альтернатива Unsplash

Сайт: pexels.com

Особенности: Огромная коллекция, есть русский интерфейс

Лицензия: Бесплатно для коммерческого использования

Бонус: Можно искать на русском языке

Пример поиска:

"сад огород"
"цветы розы"
"книги чтение"
"готовка еда"

3. Pixabay — разнообразный контент

Сайт: pixabay.com

Особенности: Фотографии, иллюстрации, векторы

Лицензия: Бесплатно, с некоторыми ограничениями

Язык: Есть русская версия

Часть 2: Специализированные ресурсы

Для иконок и небольших изображений:

Flaticon (flaticon.com) — иконки в разных стилях

Icons8 (icons8.com) — стильные иконки и иллюстрации

Feather Icons (feathericons.com) — минималистичные иконки

Для текстур и фонов:

Subtle Patterns (subtlepatterns.com) — текстуры для фона

Transparent Textures (transparenttextures.com) — прозрачные текстуры

Часть 3: Поиск с правильными фильтрами

Использование Google Images с фильтрами:

Перейдите на images.google.com
Ведите поисковый запрос (например, "сад")
Нажмите "Инструменты" → "Права на использование"
Выберите "Отметка на коммерческое использование"

Безопасные запросы для Google:

"сад дача"
"цветы розы"
"книги библиотека"
"готовка рецепты"
"рукоделие вязание"
"пожилые люди компьютер"

Часть 4: Создание собственных изображений

Простой способ: Снимки экрана

На Windows: Клавиша Print Screen или Win + Shift + S

На Mac: Cmd + Shift + 4

Сохраните в папку Вашего проекта

Создание скриншотов для сайта:

```
html

```

Часть 5: Практическое руководство по скачиванию

Шаг-за-шагом инструкция для Unsplash:

Откройте unsplash.com

В поиске введите нужную тему (например, "rose")

Выберите понравившееся фото

Нажмите кнопку "*Download*" (бесплатно)

Сохраните в папку images вашего проекта

Структура папок для организации:

```
text
Мой первый сайт/
    ├── index.html
    └── images/
        ├── nature/      # Природа и сад
        │   ├── garden.jpg
        └── flowers/
```



Часть 6: Готовые примеры URL для тренировки

Природа и сад:

```

html
<!-- Розы -->


<!-- Сад -->


<!-- Овощи -->


```

Хобби и увлечения:

```

html
<!-- Книги -->


<!-- Готовка -->
  
<!-- Рукоделие -->  

```

Люди и портреты:

```
html  
<!-- Пожилые люди за компьютером -->  
  
  
<!-- Семья -->  

```

Часть 7: Практическое задание

Создайте галерею с изображениями из бесплатных источников:

```
html  
<!DOCTYPE html>  
<html lang="ru">  
<head>  
<title>Моя тренировочная галерея</title>  
<meta charset="utf-8">  
<style>  
.gallery {  
display: grid;  
grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));  
gap: 20px;  
padding: 20px;  
}  
.photo-item {  
border: 1px solid #ddd;  
padding: 10px;  
border-radius: 8px;  
text-align: center;  
}  
.photo-item img {  
max-width: 100%;
```

```
height: auto;
border-radius: 5px;
}
.source-info {
font-size: 0.8em;
color: #666;
margin-top: 5px;
}
</style>
</head>
<body>
<h1> Тренировочная галерея изображений</h1>
<p>Все изображения взяты из бесплатных источников с открытой лицензией.</p>

<div class="gallery">

<div class="photo-item">

<p><strong>Сад мечты</strong></p>
<p class="source-info">Источник: Unsplash</p>
</div>

<!-- Изображение 2 -->
<div class="photo-item">

<p><strong>Домашняя библиотека</strong></p>
<p class="source-info">Источник: Unsplash</p>
</div>

<!-- Изображение 3 -->
<div class="photo-item">

<p><strong>Любимая кухня</strong></p>
<p class="source-info">Источник: Unsplash</p>
</div>
```

```
<!-- Изображение 4 -->
<div class="photo-item">

<p><strong>Обучение в любом возрасте</strong></p>
<p class="source-info">Источник: Unsplash</p>
</div>
</div>

<div style="margin-top: 40px; padding: 20px; background: #f0f8ff; border-radius: 10px;">
<h2>📋 Чек-лист поиска изображений</h2>
<ul>
<li>✓ Используйте сайты с открытой лицензией (Unsplash, Pexels)</li>
<li>✓ Проверяйте права использования</li>
<li>✓ Создавайте информативные alt-тексты</li>
<li>✓ Организуйте файлы в папках</li>
<li>✓ Оптимизируйте размеры изображений</li>
</ul>
</div>
</body>
</html>
```

Часть 8: Оптимизация изображений для веба

Бесплатные инструменты для оптимизации:

[TinyPNG](#) (tinypng.com) — сжатие PNG и JPEG

[Squoosh](#) (squoosh.app) — онлайн-оптимизатор от Google

[GIMP](#) — бесплатный графический редактор

Рекомендуемые настройки:

Размер: 800-1200px по ширине для фотографий

Формат: JPEG для фото, PNG для прозрачности

Качество: 70-80% для хорошего баланса качества/размера

Часть 9: Юридические аспекты — что можно и нельзя

МОЖНО:

Использовать изображения с Unsplash, Pexels, Pixabay
Использовать для личных и учебных проектов
Использовать в некоммерческих сайтах
Модифицировать изображения (если разрешено лицензией)

НЕЛЬЗЯ:

Брать изображения с Google без проверки лицензии
Использовать фотографии людей без разрешения
Использовать логотипы и торговые марки
Выдавать чужие работы за свои

Проверка лицензии:

Всегда ищите на сайте разделы:
"License" (Лицензия)
"Terms of Use" (Условия использования)
"Royalty Free" (Бесплатно)

Часть 10: Полезные советы для начинающих

Совет 1: Создайте свою коллекцию

Сохраняйте понравившиеся изображения в тематические папки:

природа_сад
хобби_увлечения
люди_портреты
еда_готовка

Совет 2: Используйте ключевые слова

При поиске используйте конкретные запросы:
Вместо "цветок" — "роза красная сад"
Вместо "книга" — "старинные книги полка"
Вместо "еда" — "домашняя выпечка пирог"

Совет 3: Начните с малого

Сначала используйте готовые URL из примеров
Потом научитесь скачивать и сохранять локально
Затем освоите оптимизацию размеров

Совет 4: Документируйте источники

Ведите простой текстовый файл с информацией:
text

garden.jpg - Unsplash, фото John Doe, лицензия CC0
books.png - Pexels, фото Jane Smith, бесплатно

Заключение

Краткий список безопасных источников:

Unsplash.com — лучший выбор для качественных фото

Pexels.com — отличная альтернатива с русским интерфейсом

Pixabay.com — разнообразный контент

Google Images — с фильтром "Права на использование"

Что Вы теперь умеете:

- Находить бесплатные изображения для обучения
- Правильно скачивать и организовывать файлы
- Соблюдать авторские права
- Создавать информативные описания
- Оптимизировать изображения для веба

Помните: Даже при использовании бесплатных изображений важно:

Указывать источник, если это требуется лицензией

Создавать качественные alt-тексты

Организовывать файлы логично

Тестировать загрузку на разных устройствах

Теперь Вы готовы наполнять свои сайты красивыми и легальными изображениями! Начните с простых упражнений и постепенно переходите к созданию полноценных галерей и иллюстрированных статей.

8.3. Как правильно сохранить картинку в свою рабочую папку

Введение: Организация — ключ к успеху

Правильное сохранение изображений так же важно, как и правильное их использование. Хорошая организация файлов поможет Вам быстро находить нужные картинки и избежать проблем с "битыми" ссылками на сайте.

Часть 1: Подготовка рабочей структуры папок

Создаем логичную структуру:

text

Мой первый сайт/

```
    └── index.html
    └── about.html
    └── contacts.html
    └── images/
        └── photos/
            ├── garden/
            │   ├── flowers/
            │   │   ├── vegetables/
            │   │   │   ├── fruits/
            │   │   │   └── family/
            │   └── hobbies/
            └── icons/
                ├── social/
                │   └── interface/
                └── backgrounds/
                    └── temp/
    └── css/
    └── style.css
```

Основная папка для всех изображений
Фотографии
Сад и огород
Цветы
Овощи
Фрукты
Семейные фото
Хобби
Иконки
Социальные сети
Элементы интерфейса
Фоновые изображения
Временные файлы

Часть 2: Пошаговая инструкция по сохранению

Шаг 1: Находим изображение в интернете

Откройте сайт с бесплатными изображениями (Unsplash, Pexels)

Найдите подходящую картинку

Откройте её в полном размере

Шаг 2: Сохраняем изображение

Способ 1: Через меню браузера

text

Правой кнопкой мыши по изображению →
"Сохранить изображение как..." →
Выбираем папку →
Сохраняем

Способ 2: Через кнопку Download (если есть)

text

Нажимаем кнопку "Download" на сайте →
Файл автоматически скачивается →
Переносим в нужную папку

Шаг 3: Выбираем правильное место

Пример для фотографии сада:

Неправильно: C:\Users\Имя\Downloads\photo.jpg

Правильно: Мой первый сайт\images\photos\garden\flowers\rose.jpg

Часть 3: Практические примеры для разных типов изображений

Пример 1: Сохраняем фото розы для страницы о саде

text

Куда сохранять: images/photos/garden/flowers/
Имя файла: red-rose.jpg
Полный путь: Мой первый сайт\images\photos\garden\flowers\red-rose.jpg

HTML-код:

html

```

```

Пример 2: Сохраняем иконку для контактов

text

Куда сохранять: images/icons/social/
Имя файла: email-icon.png
Полный путь: Мой первый сайт\images\icons\social\email-icon.png

HTML-код:

html

```

```

Пример 3: Сохраняем фоновое изображение

text

Куда сохранять: images/backgrounds/
Имя файла: garden-background.jpg
Полный путь: Мой первый сайт\images\backgrounds\garden-
background.jpg

Часть 4: Правила именования файлов

ПРАВИЛЬНЫЕ ИМЕНА:

html

```
<!-- Осмысленные и понятные -->  
  
  
  
  
<!-- С датами для версий -->  
  

```

НЕПРАВИЛЬНЫЕ ИМЕНА:

html

```
<!-- Бессмыслицеские имена -->  
  
  
  
  
<!-- С русскими буквами и пробелами -->  

```

Рекомендации по именованию:

- Только латинские буквы**
- Нижний регистр** (лучше rose.jpg чем ROSE.JPG)
- Дефисы вместо пробелов** (red-rose.jpg)
- Осмысленные названия**
- Короткие, но описательные имена**

Часть 5: Практическое задание

Создайте папку и сохраните 5 тестовых изображений:

```
html
<!DOCTYPE html>
<html lang="ru">
<head>
<title>Практика с изображениями</title>
<meta charset="utf-8">
</head>
<body>
<h1> Моя организованная коллекция изображений</h1>

<h2>Сад и огород</h2>



<h2>Хобби</h2>



<h2>Иконки</h2>


</body>
</html>
```

Структура папок после выполнения:

```
text
Мой первый сайт/
```

```
index.html
images/
photos/
  garden/
    flowers/
      rose.jpg
      vegetables/
        tomatoes.jpg
    hobbies/
      books.jpg
      cooking.jpg
  icons/
social/
email.png
phone.png
```

Часть 6: Частые ошибки и их решение

Ошибка 1: Файл не найден

```
html
<!-- Неправильно -->
![Роза](rose.jpg)<!-- Файл в другой папке -->

<!-- Правильно -->
![Роза](images/photos/garden/flowers/rose.jpg)
```

Ошибка 2: Неправильный регистр

```
html
<!-- Неправильно -->
![Роза](images/rose.jpg)
```

Ошибка 3: Русские буквы в путях

```
html
<!-- Неправильно -->
![Роза](images/my-photos/rose.jpg)
```

Ошибка 4: Пробелы в именах

```
html
<!-- Неправильно -->


<!-- Правильно -->

```

Часть 7: Инструменты для работы с изображениями

Проверка путей:

Проверьте существование файла — дважды щелкните по нему
Убедитесь в правильности пути — сравните с HTML-кодом
Проверьте расширение — .jpg, .png, .gif

Бесплатные программы:

Paint (встроен в Windows) — простейшее редактирование
GIMP — мощный бесплатный редактор
IrfanView — просмотр и базовая обработка

Часть 8: Автоматизация процесса

Создайте шаблон структуры:

Сохраните этот текст в файл folder-structure.txt:

```
text
images/
    └── photos/
        └── garden/
            ├── flowers/
            ├── vegetables/
            └── fruits/
    └── family/
    └── hobbies/
icons/
    └── social/
        └── interface/
    └── backgrounds/
    └── temp/
```

Используйте горячие клавиши:

F2 — переименовать файл

Ctrl + C / Ctrl + V — копировать/вставить

Ctrl + X — вырезать

Delete — удалить

Часть 9: Практические советы для пенсионеров

Совет 1: Начинайте с простого

Сначала создайте одну папку `images`

Сохраняйте все изображения туда

Постепенно создавайте подпапки

Совет 2: Используйте понятные имена

`text`

Лучше: `garden-rose.jpg`

Хуже: `DSC_2547.JPG`

Лучше: `grandma-cooking-pie.jpg`

Хуже: `бабушка готовит пирог.jpg`

Совет 3: Создавайте резервные копии

Копируйте папку `images` на флешку

Или используйте облако (Яндекс.Диск, Google Drive)

Совет 4: Регулярно чистите папки

Удаляйте ненужные изображения

Проверяйте, все ли файлы используются на сайте

Следите за дубликатами

Часть 10: Проверка правильности сохранения

Чек-лист после сохранения:

Файл сохранен в правильную папку

Имя файла содержит только латинские буквы

Нет пробелов в имени (использованы дефисы)

Расширение файла правильное (`.jpg, .png`)

Файл открывается двойным щелчком

Размер файла подходит для веба (обычно 100KB-1MB)

Тестирование в браузере:

Откройте HTML-файл в браузере
Проверьте, отображаются ли все изображения
Если изображение не показывается — проверьте путь
Исправьте путь и обновите страницу (F5)

Заключение

Что Вы теперь умеете:

-  Создавать логичную структуру папок
-  Правильно сохранять изображения в нужные места
-  Давать файлам осмысленные имена
-  Избегать распространенных ошибок
-  Организовывать файлы для легкого доступа

Ключевые принципы организации:

-  **Логичная структура** — похожие файлы вместе
-  **Осмысленные имена** — понятно без открытия
-  **Единый стиль** — одинаковые правила для всех файлов
-  **Иерархия** — от общего к частному
-  **Чистота** — регулярное удаление ненужного

Помните: Хорошая организация файлов сэкономит Вам массу времени в будущем. Когда Ваш сайт вырастет до десятков страниц и сотен изображений, Вы будете благодарны себе за продуманную структуру с самого начала.

Следующие шаги:

Создайте базовую структуру папок
Сохраните 3-5 тестовых изображений
Создайте простую HTML-страницу с этими изображениями
Проверьте, что все пути работают правильно
Постепенно дополняйте коллекцию
Теперь Вы готовы создавать красивые, хорошо организованные веб-сайты с правильно сохраненными изображениями!

8.4. Практика: добавляем свою фотографию или красивую картинку на главную страницу

Введение: Делаем сайт персонализированным и визуально привлекательным

Добавление изображений на главную страницу — это важный шаг, который делает Ваш сайт уникальным и запоминающимся. Давайте превратим нашу главную страницу в визуально привлекательную визитную карточку!

Часть 1: Подготовка изображений

Создаем структуру папок:

```
text
Мой первый сайт/
└── index.html
└── about.html
└── favorite-sites.html
└── images/
    └── photos/
        ├── personal/
        │   ├── my-photo.jpg      # Личные фото
        │   └── avatar.jpg        # Ваше фото
        └── content/
            ├── garden.jpg       # Аватарка
            ├── books.jpg
            └── cooking.jpg
    └── icons/
        ├── flower.png
        ├── book.png
        └── food.png
    └── backgrounds/
        └── texture.jpg          # Контентные изображения
                                # Иконки
                                # Фоновые изображения
```

Часть 2: Подготовка фотографии

Если у Вас есть цифровая фотография:

Найдите фото на компьютере или телефоне

Перенесите в папку images/photos/personal/

Переименуйте в my-photo.jpg

Если нужно сделать новое фото:

Попросите помочь родственников или друзей

**Сделайте фото на телефон или фотоаппарат
Перенесите на компьютер через USB-кабель или email
Сохраните в нужную папку**

Рекомендации к фото:

Размер: 400x400 пикселей (идеально для сайта)

Формат: JPEG (.jpg)

Содержание: Улыбающееся лицо, хорошее освещение

Фон: Нейтральный или связанный с Вашиими увлечениями

Часть 3: Полный код главной страницы с изображениями

```
html
<!DOCTYPE html>
<html lang="ru">
<head>
<title>Мой личный сайт - Главная</title>
<meta charset="utf-8">
<style>
/* Основные стили */
body {
font-family: 'Arial', sans-serif;
max-width: 1000px;
margin: 0 auto;
padding: 20px;
line-height: 1.6;
background: linear-gradient(135deg, #f5f7fa 0%, #c3cfe2 100%);
}

/* Шапка сайта */
header {
background: white;
padding: 40px;
border-radius: 15px;
box-shadow: 0 4px 20px rgba(0,0,0,0.1);
text-align: center;
margin-bottom: 30px;
position: relative;
}

/* Аватарка */
.avatar {
width: 150px;
height: 150px;
border-radius: 50%;
```

```
border: 5px solid #667eea;
margin: 0 auto 20px;
overflow: hidden;
box-shadow: 0 4px 15px rgba(0,0,0,0.2);
}
```

```
.avatar img {
width: 100%;
height: 100%;
object-fit: cover;
}
```

```
/* Навигация */
nav {
background: white;
padding: 20px;
border-radius: 10px;
box-shadow: 0 2px 10px rgba(0,0,0,0.1);
margin-bottom: 30px;
}
```

```
.nav-links {
list-style: none;
padding: 0;
margin: 0;
display: flex;
justify-content: center;
gap: 20px;
flex-wrap: wrap;
}
```

```
.nav-links a {
text-decoration: none;
color: #333;
padding: 12px 25px;
border: 2px solid #667eea;
border-radius: 25px;
transition: all 0.3s ease;
font-weight: bold;
}
```

```
.nav-links a:hover {
background: #667eea;
color: white;
transform: translateY(-2px);
```

```
}

/* Основной контент */
.main-content {
display: grid;
gap: 30px;
}

.content-section {
background: white;
padding: 30px;
border-radius: 10px;
box-shadow: 0 2px 10px rgba(0,0,0,0.1);
}

/* Карточки с увлечениями */
.hobbies-grid {
display: grid;
grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));
gap: 20px;
margin-top: 20px;
}

.hobby-card {
background: #f8f9fa;
padding: 20px;
border-radius: 10px;
text-align: center;
border-left: 4px solid #667eea;
transition: transform 0.3s ease;
}

.hobby-card:hover {
transform: translateY(-5px);
}

.hobby-icon {
width: 60px;
height: 60px;
margin: 0 auto 15px;
border-radius: 50%;
background: #667eea;
display: flex;
align-items: center;
justify-content: center;
```

```
color: white;
font-size: 24px;
}

/* Галерея изображений */
.photo-gallery {
display: grid;
grid-template-columns: repeat(auto-fit, minmax(200px, 1fr));
gap: 15px;
margin-top: 20px;
}

.gallery-item {
border-radius: 8px;
overflow: hidden;
box-shadow: 0 2px 8px rgba(0,0,0,0.1);
transition: transform 0.3s ease;
}

.gallery-item:hover {
transform: scale(1.05);
}

.gallery-item img {
width: 100%;
height: 150px;
object-fit: cover;
}

/* Подвал */
footer {
text-align: center;
margin-top: 50px;
padding: 30px;
color: #666;
border-top: 1px solid #ddd;
background: white;
border-radius: 10px;
}

/* Адаптивность */
@media (max-width: 768px) {
.nav-links {
flex-direction: column;
align-items: center;
}
```

```
}

.hobbies-grid {
grid-template-columns: 1fr;
}
}

</style>
</head>
<body>
<!-- Шапка сайта с фотографией --&gt;
&lt;header&gt;
&lt;div class="avatar"&gt;
&lt;img src="images/photos/personal/my-photo.jpg"
alt="Моя фотография - [Ваше Имя], начинающий веб-разработчик"&gt;
&lt;/div&gt;
&lt;h1&gt;🌐 Добро пожаловать на мой сайт!&lt;/h1&gt;
&lt;p&gt;Меня зовут &lt;strong&gt;Александр&lt;/strong&gt;, и я создаю свой первый веб-сайт в возрасте 65 лет!&lt;/p&gt;
&lt;p&gt;Доказываю, что учиться никогда не поздно! 🎯&lt;/p&gt;
&lt;/header&gt;

<!-- Навигация --&gt;
&lt;nav&gt;
&lt;ul class="nav-links"&gt;
&lt;li&gt;&lt;a href="index.html"&gt;🏠 Главная&lt;/a&gt;&lt;/li&gt;
&lt;li&gt;&lt;a href="about.html"&gt;👤 Обо мне&lt;/a&gt;&lt;/li&gt;
&lt;li&gt;&lt;a href="favorite-sites.html"&gt;⭐ Любимые сайты&lt;/a&gt;&lt;/li&gt;
&lt;/ul&gt;
&lt;/nav&gt;

<!-- Основной контент --&gt;
&lt;main class="main-content"&gt;
<!-- Раздел "Обо мне" --&gt;
&lt;section class="content-section"&gt;
&lt;h2&gt;👋 Рад Вас видеть!&lt;/h2&gt;
&lt;p&gt;После 40 лет работы инженером я решил освоить совершенно новую для себя сферу — веб-разработку. Это увлекательное путешествие, которое доказывает: возраст — не помеха для обучения!&lt;/p&gt;

&lt;div class="photo-gallery"&gt;
&lt;div class="gallery-item"&gt;
&lt;img src="images/photos/content/garden.jpg"
alt="Мой сад с цветущими розами и пионами"&gt;
&lt;/div&gt;
&lt;div class="gallery-item"&gt;</pre>
```

```

</div>
<div class="gallery-item">

</div>
</div>
</section>

<!-- Раздел "Мои увлечения" --&gt;
&lt;section class="content-section"&gt;
&lt;h2&gt;🎨 Мои увлечения&lt;/h2&gt;
&lt;p&gt;В свободное время я занимаюсь тем, что приносит мне радость и вдохновение:&lt;/p&gt;

&lt;div class="hobbies-grid"&gt;
&lt;div class="hobby-card"&gt;
&lt;div class="hobby-icon"&gt;🌹&lt;/div&gt;
&lt;h3&gt;Садоводство&lt;/h3&gt;
&lt;p&gt;Выращиваю розы, пионы и овощи. Мой сад — моя гордость!&lt;/p&gt;
&lt;/div&gt;

&lt;div class="hobby-card"&gt;
&lt;div class="hobby-icon"&gt;📚&lt;/div&gt;
&lt;h3&gt;Чтение&lt;/h3&gt;
&lt;p&gt;Люблю исторические романы и классическую литературу&lt;/p&gt;
&lt;/div&gt;

&lt;div class="hobby-card"&gt;
&lt;div class="hobby-icon"&gt;🍳&lt;/div&gt;
&lt;h3&gt;Кулинария&lt;/h3&gt;
&lt;p&gt;Экспериментирую с рецептами и люблю готовить для семьи&lt;/p&gt;
&lt;/div&gt;

&lt;div class="hobby-card"&gt;
&lt;div class="hobby-icon"&gt;💻&lt;/div&gt;
&lt;h3&gt;Веб-разработка&lt;/h3&gt;
&lt;p&gt;Осваиваю HTML, CSS и создаю свои первые сайты&lt;/p&gt;
&lt;/div&gt;
&lt;/div&gt;
&lt;/section&gt;

<!-- Раздел "Мой путь в IT" --&gt;
&lt;section class="content-section"&gt;</pre>
```

```
<h2>🚀 Мой путь в веб-разработку</h2>
<div style="display: flex; align-items: center; gap: 30px; flex-wrap: wrap;">
<div style="flex: 1; min-width: 300px;">

</div>
<div style="flex: 2; min-width: 300px;">
<h3>Почему я начал учиться?</h3>
<p>Всегда интересовался технологиями, но работа не позволяла
серьезно заняться программированием. Выйдя на пенсию, я наконец-то
получил возможность реализовать свою мечту!</p>

<h3>Что уже изучил:</h3>
<ul>
<li>✓ Основы HTML и создание структуры страниц</li>
<li>✓ Работа с текстом, ссылками и изображениями</li>
<li>✓ Создание навигации между страницами</li>
<li>✓ Основы CSS для оформления сайта</li>
</ul>

<p><strong>Моя цель:</strong> Создать полноценный блог о
садоводстве для таких же enthusiasts, как я!</p>
</div>
</div>
</section>

<!-- Призыв к действию -->
<section class="content-section" style="text-align: center; background: linear-
gradient(135deg, #667eea 0%, #764ba2 100%); color: white;">
<h2>💡 Хотите тоже научиться?</h2>
<p>Если Вы тоже думаете о том, чтобы освоить веб-разработку — не
сомневайтесь!</p>
<p>Посмотрите мою подборку <a href="favorite-sites.html" style="color:
#ffd700; font-weight: bold;">любимых обучающих сайтов</a>,
которые помогли мне начать этот путь.</p>
<div style="margin-top: 20px;">
<a href="favorite-sites.html" style="display: inline-block; background: white;
color: #667eea; padding: 15px 30px; border-radius: 25px; text-decoration:
none; font-weight: bold; transition: transform 0.3s ease;">
🚀 Перейти к обучающим ресурсам
</a>
</div>
</section>
```

```
</main>

<!-- Подвал -->
<footer>
<div style="display: flex; justify-content: space-around; flex-wrap: wrap; gap: 30px; margin-bottom: 20px;">
<div>
<h3>📞 Контакты</h3>
<p>Email: <a href="mailto:alexander@example.com">alexander@example.com</a></p>
<p>Телефон: <a href="tel:+79161234567">+7 (916) 123-45-67</a></p>
</div>

<div>
<h3>🌐 Быстрые ссылки</h3>
<p><a href="index.html">Главная</a></p>
<p><a href="about.html">Обо мне</a></p>
<p><a href="favorite-sites.html">Полезные ресурсы</a></p>
</div>

<div>
<h3>🌐 Социальные сети</h3>
<p>Присоединяйтесь к моему путешествию!</p>
</div>
</div>

<hr style="margin: 20px 0;">
<p>© 2024 Мой первый сайт. Создано с ❤️, упорством и верой в себя!</p>
<p><em>"Учиться никогда не поздно, а начинать — никогда не рано!"</em></p>
</footer>
</body>
</html>
```

Часть 4: Альтернативный вариант — если нет своей фотографии

Используем красивую тематическую картинку:

```
html
<!-- В шапке вместо фото -->
<header>
<div class="avatar">

```

```
</div>
<h1>🌐 Добро пожаловать на мой сайт!</h1>
<p>Меня зовут <strong>Александр</strong>, и я создаю свой первый веб-сайт!</p>
</header>
```

Или используем иконку/аватар:

```
html
<div class="avatar">

</div>
```

Часть 5: Практическое задание

Задание 1: Создайте свою версию главной страницы

Создайте папки согласно структуре выше

Добавьте свою фотографию или выберите тематическую картинку

Скачайте 3-4 дополнительные картинки для галереи

Настройте пути к изображениям в коде

Проверьте отображение в браузере

Задание 2: Персонализируйте контент

Замените в коде:

[Ваше Имя] на ваше настоящее имя

Текст "Обо мне" на Ваш реальный опыт

Изображения на Ваши настоящие увлечения

Контактную информацию на Вашу

Часть 6: Проверка и отладка

Чек-лист проверки изображений:

Все изображения отображаются корректно

Аватарка имеет круглую форму

Галерея изображений показывает все картинки

При наведении есть анимации (если добавлены)

На мобильных устройствах всё выглядит хорошо

Alt-тексты информативны и релевантны

Если изображения не показываются:

Проверьте пути:

```
html
<!-- Правильно -->


<!-- Неправильно -->

```

Проверьте имена файлов:

Нет ли опечаток?
Правильный ли регистр? (photo.jpg vs PHOTO.JPG)
Правильное ли расширение? (.jpg, .png)

Проверьте существование файлов:

Откройте папку и убедитесь, что файлы там
Попробуйте открыть файл двойным щелчком

Часть 7: Дополнительные улучшения

Добавление подписей к фото:

```
html
<div class="gallery-item">

<p class="caption">Мой сад в мае - розы в полном цвету</p>
</div>
```

Добавление эффектов при наведении:

```
CSS
.avatar img {
transition: transform 0.3s ease;
}
```

```
.avatar img:hover {
transform: scale(1.1);
}
```

Заключение

Что Вы достигли в этом уроке:

- Создали визуально привлекательную главную страницу
- Добавили персонализированную фотографию или тематическое изображение
- Организовали галерею дополнительных изображений
- Научились правильно структурировать контент с изображениями
- Создали адаптивный дизайн для разных устройств

Ключевые навыки, которые Вы применили:

-  Работа с изображениями — добавление и оформление
-  Веб-дизайн — создание привлекательного интерфейса
-  Адаптивность — работа на разных устройствах
-  Доступность — правильные alt-тексты
-  Навигация — удобные ссылки между страницами

Ваша главная страница теперь:

Персонализированная — отражает Вашу уникальность
Визуально привлекательная — приятна для просмотра
Информативная — понятно рассказывает о вас
Профессиональная — выглядит как сайт опытного разработчика
Вдохновляющая — мотивирует других начать учиться

Следующие шаги:

Покажите сайт друзьям и получите обратную связь
Добавьте больше своих фотографий и контента
Экспериментируйте с разными стилями оформления
Постепенно добавляйте новые страницы и разделы
Теперь у Вас есть полноценная, красиваая главная страница, которой не стыдно поделиться с окружающими! Это отличное начало Вашего пути в веб-разработке.

Глава 9:

Простые таблицы (для наглядности)

9.1. Создаём таблицу: теги <table>, <tr> (строка), <td> (ячейка)

Введение: Организуем данные в стройные ряды и колонки

Таблицы в HTML — это мощный инструмент для представления структурированной информации. Они идеально подходят для расписаний, прайс-листов, сравнения характеристик и любых данных, которые имеют четкую строковую и столбцовую организацию.

Часть 1: Базовая структура HTML-таблицы

Основные теги таблицы:

<table> — контейнер для всей таблицы
<tr> (table row) — строка таблицы
<td> (table data) — ячейка с данными

Простейшая таблица 2×2:

```
html
<table>
<tr>
<td>Ячейка 1</td>
<td>Ячейка 2</td>
</tr>
<tr>
<td>Ячейка 3</td>
<td>Ячейка 4</td>
</tr>
</table>
```

Результат:

Ячейка 1	Ячейка 2
Ячейка 3	Ячейка 4

Часть 2: Визуализация структуры таблицы

Как браузер "видит" таблицу:

```

text
<table>           ← НАЧАЛО ТАБЛИЦЫ
<tr>             ← ПЕРВАЯ СТРОКА
<td>А1</td>       ← ЯЧЕЙКА А1
<td>Б1</td>       ← ЯЧЕЙКА Б1
<td>В1</td>       ← ЯЧЕЙКА В1
</tr>             ← КОНЕЦ СТРОКИ
<tr>             ← ВТОРАЯ СТРОКА
<td>А2</td>       ← ЯЧЕЙКА А2
<td>Б2</td>       ← ЯЧЕЙКА Б2
<td>В2</td>       ← ЯЧЕЙКА В2
</tr>             ← КОНЕЦ СТРОКИ
</table>           ← КОНЕЦ ТАБЛИЦЫ

```

Часть 3: Практические примеры таблиц

Пример 1: Расписание полива растений

```

html
<table border="1">
<tr>
<td>Растение</td>
<td>Понедельник</td>
<td>Среда</td>
<td>Пятница</td>
</tr>
<tr>
<td>Розы</td>
<td>✓ Полить</td>
<td>✗ Не поливать</td>
<td>✓ Полить</td>
</tr>
<tr>
<td>Помидоры</td>
<td>✓ Полить</td>
<td>✓ Полить</td>
<td>✓ Полить</td>
</tr>
<tr>
<td>Кактусы</td>
<td>✗ Не поливать</td>
<td>✗ Не поливать</td>
<td>✓ Полить</td>
</tr>
</table>

```

Результат:

Растение	Понедельник	Среда	Пятница
Розы	<input checked="" type="checkbox"/> Полить	<input checked="" type="checkbox"/> Не поливать	<input checked="" type="checkbox"/> Полить
Помидоры	<input checked="" type="checkbox"/> Полить	<input checked="" type="checkbox"/> Полить	<input checked="" type="checkbox"/> Полить
Кактусы	<input checked="" type="checkbox"/> Не поливать	<input checked="" type="checkbox"/> Не поливать	<input checked="" type="checkbox"/> Полить

Пример 2: Список покупок для сада

```
html
<table border="1">
<tr>
<td>№</td>
<td>Товар</td>
<td>Количество</td>
<td>Цена</td>
</tr>
<tr>
<td>1</td>
<td>Удобрение для роз</td>
<td>2 шт.</td>
<td>350 руб.</td>
</tr>
<tr>
<td>2</td>
<td>Семена помидоров</td>
<td>1 упаковка</td>
<td>120 руб.</td>
</tr>
<tr>
<td>3</td>
<td>Садовые перчатки</td>
<td>1 пара</td>
<td>280 руб.</td>
</tr>
</table>
```

Пример 3: График занятий HTML

```
html
```

```
<table border="1">
<tr>
<td>День недели</td>
<td>Тема занятия</td>
<td>Время</td>
</tr>
<tr>
<td>Понедельник</td>
<td>Изучение тегов</td>
<td>14:00 - 15:30</td>
</tr>
<tr>
<td>Среда</td>
<td>Работа с текстом</td>
<td>14:00 - 16:00</td>
</tr>
<tr>
<td>Пятница</td>
<td>Практика: создание страниц</td>
<td>15:00 - 17:00</td>
</tr>
</table>
```

Часть 4: Атрибут border — границы таблицы

Без границ:

```
html
<table>
<!-- Браузер не показывает границы --&gt;
&lt;/table&gt;</pre>
```

С тонкой границей:

```
html
<table border="1">
<!-- Тонкая граница вокруг таблицы и ячеек --&gt;
&lt;/table&gt;</pre>
```

С толстой границей:

```
html
<table border="5">
<!-- Более толстая граница --&gt;
&lt;/table&gt;</pre>
```

Важно: Атрибут border устарел, но идеально подходит для обучения, так как сразу делает таблицу видимой.

Часть 5: Полная страница с различными таблицами

```
html
<!DOCTYPE html>
<html lang="ru">
<head>
<title>Практика с таблицами</title>
<meta charset="utf-8">
<style>
body {
font-family: Arial, sans-serif;
max-width: 900px;
margin: 0 auto;
padding: 20px;
line-height: 1.6;
}
.table-container {
margin: 30px 0;
padding: 20px;
border: 1px solid #ddd;
border-radius: 10px;
}
table {
width: 100%;
border-collapse: collapse;
margin: 15px 0;
}
th, td {
border: 1px solid #ccc;
padding: 12px;
text-align: left;
}
th {
background-color: #f2f2f2;
font-weight: bold;
}
.garden-schedule tr:nth-child(even) {
background-color: #f9f9f9;
}
.shopping-list td:first-child {
font-weight: bold;
```

```
background-color: #e6f3ff;
}
</style>
</head>
<body>
<h1> Практика работы с HTML-таблицами</h1>

<!-- Таблица 1: Расписание ухода за садом --&gt;
<div class="table-container">


## <img alt="Flower icon" data-bbox="155 245 175 265"/> График ухода за садом</h2> | День недели | Утренние работы<br>(8:00 - 10:00) | Вечерние работы<br>(18:00 - 20:00) | Примечания | |-------------|-----------------------------------|------------------------------------|-----------------------------------------| | Понедельник | Полив цветов | Прополка грядок | Проверить розы на вредителей | | Вторник | Подкормка овощей | Рыхление почвы | Удобрить помидоры | | Среда | Обрезка кустов | Полив всего сада | Особенно тщательно полить новые посадки | | Четверг | Сбор урожая | Подвязка растений | Собрать спелые огурцы и помидоры | | Пятница | Уход за газоном | | Подготовка к выходным |


```

```
<td>Проверить весь инвентарь</td>
</tr>
</table>
</div>

<!-- Таблица 2: Список растений --&gt;
<div class="table-container">


## Мои растения в саду



| Название растения       | Дата посадки | Местоположение    | Особенности ухода                          |
|-------------------------|--------------|-------------------|--------------------------------------------|
| Роза "Красная сказка"   | 15.04.2024   | Солнечная сторона | Полив через день, подкормка раз в 2 недели |
| Помидоры "Бычье сердце" | 20.05.2024   | Теплица           | Ежедневный полив, пасынкование             |
| Огурцы "Зозуля"         | 18.05.2024   | Грядка № 3        | Обильный полив, сбор каждые 2 дня          |
| Пионы "Сара Бернар"     | 10.09.2023   | Цветник у дома    | Минимальный уход, подкормка весной         |


```

```
<!-- Таблица 3: План изучения HTML --&gt;
<div class="table-container">


## Мой план изучения HTML


```

```

<table>
<tr>
<th>Неделя</th>
<th>Тема</th>
<th>Что изучил</th>
<th>Статус</th>
</tr>
<tr>
<td>1</td>
<td>Основы HTML</td>
<td>Структура страницы, теги, атрибуты</td>
<td>✓ Завершено</td>
</tr>
<tr>
<td>2</td>
<td>Работа с текстом</td>
<td>Заголовки, абзацы, списки, форматирование</td>
<td>✓ Завершено</td>
</tr>
<tr>
<td>3</td>
<td>Ссылки и изображения</td>
<td>Гиперссылки, картинки, навигация</td>
<td>✓ Завершено</td>
</tr>
<tr>
<td>4</td>
<td>Таблицы</td>
<td>Создание таблиц, строки, ячейки</td>
<td>⌚ В процессе</td>
</tr>
<tr>
<td>5</td>
<td>Формы</td>
<td>Поля ввода, кнопки, отправка данных</td>
<td>⏳ В планах</td>
</tr>
</table>
</div>

```

```

<!-- Таблица 4: Простая таблица для начинающих --&gt;
&lt;div class="table-container"&gt;
&lt;h2&gt;🎯 Простая таблица для тренировки&lt;/h2&gt;
&lt;table border="1"&gt;
&lt;tr&gt;
</pre>

```

```
<td>Ячейка A1</td>
<td>Ячейка A2</td>
<td>Ячейка A3</td>
</tr>
<tr>
<td>Ячейка B1</td>
<td>Ячейка B2</td>
<td>Ячейка B3</td>
</tr>
<tr>
<td>Ячейка C1</td>
<td>Ячейка C2</td>
<td>Ячейка C3</td>
</tr>
</table>
<p><em>Попробуйте изменить содержимое ячеек!</em></p>
</div>
</body>
</html>
```

Часть 6: Объединение ячеек

Объединение по горизонтали (colspan):

```
html
<table border="1">
<tr>
<td colspan="2">Эта ячейка занимает две колонки</td>
</tr>
<tr>
<td>Ячейка 1</td>
<td>Ячейка 2</td>
</tr>
</table>
```

Объединение по вертикали (rowspan):

```
html
<table border="1">
<tr>
<td rowspan="2">Эта ячейка занимает две строки</td>
<td>Ячейка 1</td>
</tr>
<tr>
<td>Ячейка 2</td>
```

```
</tr>
</table>
```

Часть 7: Практическое задание

Задание 1: Создайте таблицу распорядка дня

```
html
<table border="1">
<tr>
<th>Время</th>
<th>Понедельник</th>
<th>Вторник</th>
<th>Среда</th>
</tr>
<tr>
<td>8:00 - 10:00</td>
<td>Зарядка, завтрак</td>
<td>Зарядка, завтрак</td>
<td>Зарядка, завтрак</td>
</tr>
<tr>
<td>10:00 - 12:00</td>
<td>Работа в саду</td>
<td>Изучение HTML</td>
<td>Работа в саду</td>
</tr>
<!-- Добавьте свои занятия --&gt;
&lt;/table&gt;</pre>
```

Задание 2: Таблица любимых книг

```
html
<table border="1">
<tr>
<th>Автор</th>
<th>Название книги</th>
<th>Год прочтения</th>
<th>Оценка</th>
</tr>
<tr>
<td>Александр Дюма</td>
<td>Три мушкетера</td>
<td>2023</td>
<td>★★★★★</td>
```

```
</tr>
<!-- Добавьте свои любимые книги --&gt;
&lt;/table&gt;</pre>
```

Часть 8: Частые ошибки и лучшие практики

НЕПРАВИЛЬНО:

```
html
<!-- Неправильная вложенность --&gt;
<table>
<td>Ячейка без строки</td> <!-- Должен быть внутри tr -->
</table>

<!-- Не закрыты теги --&gt;
<table&gt;
&lt;tr&gt;
&lt;td&gt;Ячейка 1
&lt;td&gt;Ячейка 2 &lt;!-- Нет закрывающих &lt;/td&gt; --&gt;
&lt;/tr&gt;
&lt;/table&gt;</pre>
```

ПРАВИЛЬНО:

```
html
<!-- Правильная структура --&gt;
<table&gt;
&lt;tr&gt;
&lt;td&gt;Ячейка 1&lt;/td&gt;
&lt;td&gt;Ячейка 2&lt;/td&gt;
&lt;/tr&gt;
&lt;/table&gt;

<!-- С семантическими тегами --&gt;
<table&gt;
&lt;tr&gt;
&lt;th&gt;Заголовок 1&lt;/th&gt;
&lt;th&gt;Заголовок 2&lt;/th&gt;
&lt;/tr&gt;
&lt;tr&gt;
&lt;td&gt;Данные 1&lt;/td&gt;
&lt;td&gt;Данные 2&lt;/td&gt;
&lt;/tr&gt;
&lt;/table&gt;</pre>
```

Часть 9: Семантика и доступность

Тег <th> для заголовков:

```
html
<table>
<tr>
<th>Название</th>    <!-- th вместо td для заголовков --&gt;
&lt;th&gt;Цена&lt;/th&gt;
&lt;th&gt;Количество&lt;/th&gt;
&lt;/tr&gt;
&lt;tr&gt;
&lt;td&gt;Помидоры&lt;/td&gt;
&lt;td&gt;150 руб/кг&lt;/td&gt;
&lt;td&gt;2 кг&lt;/td&gt;
&lt;/tr&gt;
&lt;/table&gt;</pre>
```

Scope для скринридеров:

```
html
<table>
<tr>
<th scope="col">Продукт</th>    <!-- Заголовок колонки --&gt;
&lt;th scope="col"&gt;Цена&lt;/th&gt;
&lt;/tr&gt;
&lt;tr&gt;
&lt;th scope="row"&gt;Хлеб&lt;/th&gt;        <!-- Заголовок строки --&gt;
&lt;td&gt;50 руб.&lt;/td&gt;
&lt;/tr&gt;
&lt;/table&gt;</pre>
```

Заключение

Что Вы теперь умеете:

- Создавать базовые HTML-таблицы
- Использовать теги <table>, <tr>, <td>
- Добавлять границы с помощью атрибута border
- Создавать сложные таблицы с заголовками
- Организовывать данные в стройные структуры

Ключевые преимущества таблиц:

-  **Структурированность** — четкая организация данных

-  **Сравнимость** — легко сравнивать значения
-  **Наглядность** — информация воспринимается легче
-  **Доступность** — правильная семантика для скринридеров
-  **Адаптивность** — можно сделать отзывчивой

Когда использовать таблицы:

-  Табличные данные (расписания, прайс-листы)
-  Сравнительные характеристики
-  Статистическая информация
-  Структурированные списки

Когда НЕ использовать таблицы:

-  Для верстки страницы (используйте CSS)
-  Для простого расположения элементов
-  Когда данные не имеют табличной природы

Следующие шаги:

Попрактикуйтесь с простыми таблицами 2×2 и 3×3

Создайте таблицу своего недельного расписания

Добавьте таблицу на свою главную страницу

Экспериментируйте с объединением ячеек

Теперь Вы можете создавать структурированные таблицы для организации любой информации на Вашем сайте! Это особенно полезно для расписаний, списков, сравнений и других табличных данных.

9.2. Заголовки столбцов в таблице: тег <th>

Введение: Делаем таблицы понятными и семантически правильными

Тег **<th>** (table header) — это специальный тег для создания заголовков в таблицах. Он не только визуально выделяет заголовки, но и добавляет семантический смысл, помогая браузерам, поисковым системам и скринридерам правильно понимать структуру ваших данных.

Часть 1: Что такое тег <th> и зачем он нужен?

Основное назначение:

<th> — создает ячейку-заголовок для столбца или строки
<td> — создает обычную ячейку с данными

Визуальная разница:

По умолчанию браузер отображает <th>:

Жирным шрифтом
Выравниванием по центру
Семантически как заголовок

Часть 2: Сравнение <th> и <td>

Без <th> (неправильно):

```
html
<table border="1">
<tr>
<td><strong>Растение</strong></td>    <!-- Приходится использовать
strong --&gt;
&lt;td&gt;&lt;strong&gt;Полив&lt;/strong&gt;&lt;/td&gt;
&lt;td&gt;&lt;strong&gt;Удобрение&lt;/strong&gt;&lt;/td&gt;
&lt;/tr&gt;
&lt;tr&gt;
&lt;td&gt;Розы&lt;/td&gt;
&lt;td&gt;Каждый день&lt;/td&gt;
&lt;td&gt;Раз в неделю&lt;/td&gt;
&lt;/tr&gt;
&lt;/table&gt;</pre>
```

С <th> (правильно):

```

html
<table border="1">
<tr>
<th>Растение</th>    <!-- Браузер сам сделает жирным --&gt;
&lt;th&gt;Полив&lt;/th&gt;
&lt;th&gt;Удобрение&lt;/th&gt;
&lt;/tr&gt;
&lt;tr&gt;
&lt;td&gt;Розы&lt;/td&gt;
&lt;td&gt;Каждый день&lt;/td&gt;
&lt;td&gt;Раз в неделю&lt;/td&gt;
&lt;/tr&gt;
&lt;/table&gt;
</pre>

```

Результат:

Растение	Полив	Удобрение
Розы	Каждый день	Раз в неделю

Часть 3: Практические примеры с <th>

Пример 1: Таблица ухода за растениями

```

html
<table border="1">
<tr>
<th>Название растения</th>
<th>Частота полива</th>
<th>Лучшее время для полива</th>
<th>Особые указания</th>
</tr>
<tr>
<td>Розы</td>
<td>Каждый день</td>
<td>Утро (6:00-8:00)</td>
<td>Не поливать листья</td>
</tr>
<tr>
<td>Помидоры</td>
<td>Через день</td>
<td>Вечер (18:00-20:00)</td>

```

```
<td>Поливать под корень</td>
</tr>
<tr>
<td>Кактусы</td>
<td>Раз в неделю</td>
<td>Утро</td>
<td>Маленькими порциями</td>
</tr>
</table>
```

Пример 2: Расписание занятий

```
html
<table border="1">
<tr>
<th>Время</th>
<th>Понедельник</th>
<th>Среда</th>
<th>Пятница</th>
</tr>
<tr>
<td>10:00-11:30</td>
<td>HTML: основы</td>
<td>CSS: стили</td>
<td>Практика</td>
</tr>
<tr>
<td>14:00-15:30</td>
<td>Садоводство</td>
<td>Чтение</td>
<td>Отдых</td>
</tr>
</table>
```

Пример 3: Список покупок для сада

```
html
<table border="1">
<tr>
<th>№</th>
<th>Наименование</th>
<th>Количество</th>
<th>Примерная цена</th>
<th>Приоритет</th>
</tr>
```

```

<tr>
<td>1</td>
<td>Удобрение для роз</td>
<td>2 кг</td>
<td>450 руб.</td>
<td>Высокий</td>
</tr>
<tr>
<td>2</td>
<td>Семена огурцов</td>
<td>1 упаковка</td>
<td>85 руб.</td>
<td>Средний</td>
</tr>
<tr>
<td>3</td>
<td>Садовые перчатки</td>
<td>1 пара</td>
<td>320 руб.</td>
<td>Низкий</td>
</tr>
</table>

```

Часть 4: Атрибут **scope** — для улучшения доступности

Что такое **scope**:

Атрибут **scope** указывает, к чему относится заголовок — к столбцу или строке.

scope="col" — заголовок столбца:

```

html
<table border="1">
<tr>
<th scope="col">Растение</th>    <!-- Заголовок для всего столбца -->
<th scope="col">Январь</th>
<th scope="col">Февраль</th>
<th scope="col">Март</th>
</tr>
<tr>
<td>Розы</td>
<td>Обрезка</td>
<td>Укрытие</td>
<td>Пробуждение</td>

```

```
</tr>
</table>
```

scope="row" — заголовок строки:

```
html


| Месяц   | Работы в саду             |
|---------|---------------------------|
| Январь  | Планирование, заказ семян |
| Февраль | Посев рассады             |


```

Часть 5: Полная страница с семантическими таблицами

```
html
!DOCTYPE html>


Семантические таблицы с тегом<th>


```

```
table {
width: 100%;
border-collapse: collapse;
margin: 20px 0;
font-size: 14px;
}
th, td {
border: 1px solid #ddd;
padding: 12px;
text-align: left;
}
th {
background-color: #4CAF50;
color: white;
font-weight: bold;
}
tr:nth-child(even) {
background-color: #f9f9f9;
}
tr:hover {
background-color: #f1f1f1;
}
.note {
background: #e7f3ff;
padding: 15px;
border-left: 4px solid #2196F3;
margin: 15px 0;
border-radius: 5px;
}
.code-example {
background: #2d2d2d;
color: #f8f8f2;
padding: 15px;
border-radius: 5px;
font-family: 'Courier New', monospace;
margin: 15px 0;
overflow-x: auto;
}

```

</style>

</head>

<body>

<h1>  Заголовки таблиц: тег<th></h1>

<div class="note">

 Важно: Тег<th> не только визуально выделяет заголовки, но и добавляет семантический смысл для браузеров и скринридеров.

</div>

```
<!-- Пример 1: Календарь садовых работ -->
<div class="table-section">
<h2> Годовой календарь садовода</h2>

<table>
<tr>
<th scope="col">Сезон</th>
<th scope="col">Основные работы</th>
<th scope="col">Посадка</th>
<th scope="col">Уход</th>
<th scope="col">Сбор урожая</th>
</tr>
<tr>
<th scope="row">Весна<br>(Март-Май)</th>
<td>Подготовка почвы, посев</td>
<td>Овощи, цветы</td>
<td>Полив, прополка</td>
<td>Редис, салат</td>
</tr>
<tr>
<th scope="row">Лето<br>(Июнь-Август)</th>
<td>Уход за растениями</td>
<td>Поздние культуры</td>
<td>Подкормка, защита</td>
<td>Огурцы, помидоры</td>
</tr>
<tr>
<th scope="row">Осень<br>(Сентябрь-Ноябрь)</th>
<td>Уборка, подготовка к зиме</td>
<td>Озимые культуры</td>
<td>Обрезка, укрытие</td>
<td>Яблоки, груши</td>
</tr>
<tr>
<th scope="row">Зима<br>(Декабрь-Февраль)</th>
<td>Планирование, ремонт</td>
<td>Рассада в доме</td>
<td>Защита от морозов</td>
<td>Хранение урожая</td>
</tr>
```

```
</table>
</div>

<!-- Пример 2: Сравнение сортов растений --&gt;
<div class="table-section">


## Сравнение сортов помидоров



| Сорт         | Срок созревания | Вкус          | Урожайность   | Устойчивость к болезням | Рекомендации               |
|--------------|-----------------|---------------|---------------|-------------------------|----------------------------|
| Бычье сердце | 120-130 дней    | Отличный      | Средняя       | Средняя                 | Для салатов, крупные плоды |
| Черри        | 90-100 дней     | Сладкий       | Высокая       | Высокая                 | Для закусок, консервации   |
| De Барао     | 115-125 дней    | Кисло-сладкий | Очень высокая | Высокая                 | Универсальный, для всего   |


```

```
<!-- Пример 3: План изучения веб-технологий --&gt;
<div class="table-section">
```

<h2>💻 План обучения веб-разработке</h2>

```
<table>
<tr>
<th scope="col">Этап</th>
<th scope="col">Технология</th>
<th scope="col">Срок изучения</th>
<th scope="col">Что изучим</th>
<th scope="col">Практический проект</th>
<th scope="col">Статус</th>
</tr>
<tr>
<th scope="row">1</th>
<td>HTML</td>
<td>3-4 недели</td>
<td>Структура, теги, семантика</td>
<td>Личный сайт-визитка</td>
<td>✅ Завершено</td>
</tr>
<tr>
<th scope="row">2</th>
<td>CSS</td>
<td>4-5 недель</td>
<td>Стили, макет, анимации</td>
<td>Красивое оформление сайта</td>
<td>⌚ В процессе</td>
</tr>
<tr>
<th scope="row">3</th>
<td>JavaScript</td>
<td>6-8 недель</td>
<td>Интерактивность, логика</td>
<td>Интерактивные элементы</td>
<td>⏳ В планах</td>
</tr>
</table>
</div>
```

```
<!-- Пример кода -->
<div class="table-section">
<h2>📝 Пример кода с<th></h2>
```

```
<div class="code-example">
<table border="1">
<tr>
```

```

<th scope="col">Растение</th>
<th scope="col">Полив</th>
<th scope="col">Удобрение</th>
</tr>
<tr>
<td>Розы</td>
<td>Каждый день</td>
<td>Раз в неделю</td>
</tr>
</table>
</div>

<div class="note">
<strong>🎯 Ключевые моменты:</strong>
<ul>
<li>Используйте <code><th></code> для заголовков</li>
<li>Добавляйте <code>scope="col"</code> для заголовков столбцов</li>
<li>Добавляйте <code>scope="row"</code> для заголовков строк</li>
<li>Это улучшает доступность для скринридеров</li>
</ul>
</div>
</div>


<div class="table-section">
<h2>🎯 Практическое задание</h2>

<h3>Создайте таблицу своих ежедневных дел:</h3>

<table border="1">
<tr>
<th scope="col">Время</th>
<th scope="col">Понедельник</th>
<th scope="col">Вторник</th>
<th scope="col">Среда</th>
</tr>
<tr>
<th scope="row">8:00-10:00</th>
<td>Завтрак, зарядка</td>
<td>Завтрак, зарядка</td>
<td>Завтрак, зарядка</td>
</tr>
<tr>
<th scope="row">10:00-12:00</th>
<td>▼ Ваши занятия</td>

```

```

<td>▼ Ваши занятия</td>
<td>▼ Ваши занятия</td>
</tr>
<tr>
<th scope="row">14:00-16:00</th>
<td>▼ Ваши занятия</td>
<td>▼ Ваши занятия</td>
<td>▼ Ваши занятия</td>
</tr>
</table>

<p><strong>Задание:</strong> Замените "▼ Ваши занятия" на свои реальные дела!</p>
</div>
</body>
</html>

```

Часть 6: Стилизация заголовков <th>

Базовые CSS-стили:

```

css
th {
background-color: #4CAF50; /* Зеленый фон */
color: white; /* Белый текст */
font-weight: bold; /* Жирный шрифт */
text-align: center; /* Выравнивание по центру */
padding: 12px; /* Внутренние отступы */
border: 1px solid #ddd; /* Граница */
}

```

Разные цвета для разных типов заголовков:

```

css
/* Заголовки столбцов */
th[scope="col"] {
background-color: #2196F3; /* Синий */
}

/* Заголовки строк */
th[scope="row"] {
background-color: #FF9800; /* Оранжевый */
text-align: left;
}

```

Часть 7: Лучшие практики и частые ошибки

ПРАВИЛЬНО:

```
html
<!-- Семантически правильная таблица -->
<table>
<tr>
<th scope="col">Название</th>
<th scope="col">Цена</th>
<th scope="col">Количество</th>
</tr>
<tr>
<th scope="row">Помидоры</th>
<td>150 руб/кг</td>
<td>2 кг</td>
</tr>
</table>
```

НЕПРАВИЛЬНО:

```
html
<!-- Смешивание th и td без смысла -->
<table>
<tr>
<td><strong>Название</strong></td> <!-- Используйте th! -->
<th>Цена</th>
<td>Количество</td>
</tr>
</table>

<!-- Отсутствие scope -->
<table>
<tr>
<th>Название</th> <!-- Добавьте scope="col" -->
<th>Цена</th>
</tr>
</table>
```

Часть 8: Доступность для скринридеров

Как скринридеры читают таблицы:

Без `<th>`: "Таблица с 3 столбцами и 4 строками"

C <th>: "Таблица: Растение, Полив, Удобрение. Стока 1: Розы, Каждый день, Раз в неделю"

Правильная структура для доступности:

```
html
<table>
<tr>
<th scope="col" id="plant">Растение</th>
<th scope="col" id="water">Полив</th>
<th scope="col" id="fertilize">Удобрение</th>
</tr>
<tr>
<th scope="row" id="roses">Розы</th>
<td headers="plant roses water">Каждый день</td>
<td headers="plant roses fertilize">Раз в неделю</td>
</tr>
</table>
```

Часть 9: Практическое задание для закрепления

Задание: Создайте таблицу ваших увлечений

```
html
<table border="1">
<tr>
<th scope="col">Увлечение</th>
<th scope="col">Как часто занимаюсь</th>
<th scope="col">Любимый аспект</th>
<th scope="col">Что планирую изучить</th>
</tr>
<tr>
<th scope="row">Садоводство</th>
<td>Ежедневно</td>
<td>Выращивание роз</td>
<td>Новые сорта пионов</td>
</tr>
<tr>
<th scope="row">Веб-разработка</th>
<td>3 раза в неделю</td>
<td>Создание сайтов</td>
<td>CSS и JavaScript</td>
</tr>
<tr>
<th scope="row">Чтение</th>
```

```
<td>Каждый вечер</td>
<td>Исторические романы</td>
<td>Современную литературу</td>
</tr>
</table>
```

Заключение

Что Вы теперь умеете:

- Использовать тег `<th>` для заголовков таблиц
- Различать `<th>` и `<td>` по назначению
- Применять атрибут `scope` для доступности
- Создавать семантически правильные таблицы
- Стилизовать заголовки для лучшего восприятия

Ключевые преимущества `<th>`:

-  **Доступность** — помочь скринридерам
-  **SEO** — лучшая индексация поисковыми системами
-  **Визуальное выделение** — понятная структура
-  **Семантика** — ясное назначение элементов
-  **Единообразие** — стандартное поведение в браузерах

Золотые правила использования `<th>`:

- Всегда используйте `<th>` для заголовков**
- Добавляйте `scope="col"` для заголовков столбцов**
- Добавляйте `scope="row"` для заголовков строк**
- Используйте осмысленные и понятные заголовки**
- Тестируйте со скринридерами для проверки доступности**

Помните: Правильное использование `<th>` превращает Вашу таблицу из простого набора данных в хорошо структурированную, доступную и профессиональную информацию!

Теперь Ваши таблицы будут не только красивыми, но и семантически правильными, что особенно важно для пользователей с ограниченными возможностями и для поисковых систем.

9.3. Простая рамка для таблицы

Введение: Делаем таблицы видимыми и аккуратными

Рамка вокруг таблицы — это не просто декоративный элемент. Она помогает визуально отделить таблицу от остального контента, делает данные более читаемыми и создает четкую структуру. Давайте научимся управлять рамками таблиц!

Часть 1: Базовые способы создания рамки

Способ 1: Атрибут `border` (самый простой)

html

```
<table border="1">
<tr>
<th>Растение</th>
<th>Полив</th>
</tr>
<tr>
<td>Розы</td>
<td>Каждый день</td>
</tr>
</table>
```

Результат:

Растение	Полив
Розы	Каждый день

Разные значения `border`:

html

```
<table border="1"> <!-- Тонкая рамка -->
<table border="3"> <!-- Средняя рамка -->
<table border="5"> <!-- Толстая рамка -->
<table border="0"> <!-- Без рамки -->
```

Часть 2: CSS-стили для продвинутого оформления

Способ 2: CSS свойство border

html

```
<table style="border: 2px solid black;">  
  <!-- Содержимое таблицы -->  
</table>
```

Разбор CSS border:

CSS

```
border: 2px solid black;  
/*  
 2px - толщина рамки  
solid - стиль линии (сплошная)  
black - цвет рамки  
*/
```

Часть 3: Различные стили рамок

Сплошная рамка (solid):

html

```
<table style="border: 3px solid #3366cc;">  
<table style="border: 3px solid #3366cc; margin: 15px 0;">  
<tr>  
<td style="padding: 10px;">  
  Сплошная синяя рамка  
</td>  
</tr>  
</table>
```

Пунктирная рамка (dashed):

html

```
<table style="border: 2px dashed #ff6600;">  
<table style="border: 2px dashed #ff6600; margin: 15px 0;">  
<tr>  
<td style="padding: 10px;">  
  Пунктирная оранжевая рамка  
</td>  
</tr>  
</table>
```

Точечная рамка (dotted):

```
html
<table style="border: 3px dotted #009933;">
<table style="border: 3px dotted #009933; margin: 15px 0;">
<tr>
<td style="padding: 10px;">
Точечная зеленая рамка
</td>
</tr>
</table>
```

Двойная рамка (double):

```
html
<table style="border: 4px double #9933cc;">
<table style="border: 4px double #9933cc; margin: 15px 0;"> <tr><td
style="padding: 10px;">Двойная фиолетовая рамка</td></tr> </table>
```

Часть 4: Полная страница с примерами рамок

```
html
<!DOCTYPE html>
<html lang="ru">
<head>
<title>Рамки для таблиц</title>
<meta charset="utf-8">
<style>
body {
font-family: 'Arial', sans-serif;
max-width: 1000px;
margin: 0 auto;
padding: 20px;
line-height: 1.6;
background-color: #f8f9fa;
}
.example-section {
background: white;
padding: 25px;
margin: 25px 0;
border-radius: 10px;
box-shadow: 0 2px 5px rgba(0,0,0,0.1);
}
h2 {
```

```
color: #2c3e50;
border-bottom: 2px solid #3498db;
padding-bottom: 10px;
}
.code-example {
background: #2d2d2d;
color: #f8f8f2;
padding: 15px;
border-radius: 5px;
font-family: 'Courier New', monospace;
margin: 15px 0;
overflow-x: auto;
}
.note {
background: #e7f3ff;
padding: 15px;
border-left: 4px solid #2196F3;
margin: 15px 0;
border-radius: 5px;
}
/* Стили для демонстрационных таблиц */
.demo-table {
margin: 20px 0;
width: 100%;
}
.demo-table th, .demo-table td {
padding: 10px;
text-align: center;
}
</style>
</head>
<body>
<h1> Рамки для HTML-таблиц</h1>

<div class="note">
<strong> Совет:</strong> Рамка делает таблицу более читаемой и визуально привлекательной. Выбирайте стиль, который лучше всего подходит для Вашего контента!
</div>

<!-- Пример 1: Простая рамка через атрибут -->
<div class="example-section">
<h2>1. Простая рамка через атрибут border</h2>

<div class="code-example">
```

```
<table border="1">
<tr>
<th>Растение</th>
<th>Полив</th>
</tr>
<tr>
<td>Розы</td>
<td>Каждый день</td>
</tr>
</table>
</div>

<table border="1" class="demo-table">
<tr>
<th>Растение</th>
<th>Полив</th>
<th>Удобрение</th>
</tr>
<tr>
<td>Розы</td>
<td>Каждый день</td>
<td>Раз в неделю</td>
</tr>
<tr>
<td>Помидоры</td>
<td>Через день</td>
<td>Раз в 2 недели</td>
</tr>
</table>
</div>

<!-- Пример 2: CSS рамки --&gt;
&lt;div class="example-section"&gt;
&lt;h2&gt;2. CSS-рамки с разными стилями&lt;/h2&gt;

&lt;!-- Сплошная рамка --&gt;
&lt;h3&gt;Сплошная рамка (solid)&lt;/h3&gt;
&lt;div class="code-example"&gt;
&lt;table style="border: 2px solid #3498db;"&gt;
&lt;/div&gt;
&lt;table style="border: 2px solid #3498db;" class="demo-table"&gt;
&lt;tr&gt;&lt;td&gt;Синяя сплошная рамка&lt;/td&gt;&lt;/tr&gt;
&lt;/table&gt;

&lt;!-- Пунктирная рамка --&gt;</pre>
```

```

<h3>Пунктирная рамка (dashed)</h3>
<div class="code-example">
<table style="border: 3px dashed #e74c3c;">
</div>
<table style="border: 3px dashed #e74c3c;" class="demo-table">
<tr><td>Красная пунктирная рамка</td></tr>
</table>

<!-- Точечная рамка -->
<h3>Точечная рамка (dotted)</h3>
<div class="code-example">
<table style="border: 3px dotted #27ae60;">
</div>
<table style="border: 3px dotted #27ae60;" class="demo-table">
<tr><td>Зеленая точечная рамка</td></tr>
</table>

<!-- Двойная рамка -->
<h3>Двойная рамка (double)</h3>
<div class="code-example">
<table style="border: 5px double #8e44ad;">
</div>
<table style="border: 5px double #8e44ad;" class="demo-table">
<tr><td>Фиолетовая двойная рамка</td></tr>
</table>
</div>

<!-- Пример 3: Практическая таблица с рамкой -->
<div class="example-section">
<h2>3. Практический пример: График полива растений</h2>

<table style="border: 3px solid #2c3e50; border-collapse: collapse; width: 100%;">
<tr style="background-color: #34495e; color: white;">
<th style="border: 1px solid #2c3e50; padding: 12px;">Растение</th>
<th style="border: 1px solid #2c3e50; padding: 12px;">Понедельник</th>
<th style="border: 1px solid #2c3e50; padding: 12px;">Среда</th>
<th style="border: 1px solid #2c3e50; padding: 12px;">Пятница</th>
<th style="border: 1px solid #2c3e50; padding: 12px;">Воскресенье</th>
</tr>
<tr>
<td style="border: 1px solid #bdc3c7; padding: 10px; background: #ecf0f1;">Розы</td>
<td style="border: 1px solid #bdc3c7; padding: 10px;">☑ Полить</td>
<td style="border: 1px solid #bdc3c7; padding: 10px;">☒ Не поливать</td>

```

```

<td style="border: 1px solid #bdc3c7; padding: 10px;">✓ Полить</td>
<td style="border: 1px solid #bdc3c7; padding: 10px;">✓ Полить</td>
</tr>
<tr>
<td style="border: 1px solid #bdc3c7; padding: 10px; background:
#ecf0f1;">Помидоры</td>
<td style="border: 1px solid #bdc3c7; padding: 10px;">✓ Полить</td>
</tr>
<tr>
<td style="border: 1px solid #bdc3c7; padding: 10px; background:
#ecf0f1;">Кактусы</td>
<td style="border: 1px solid #bdc3c7; padding: 10px;">✗ Не поливать</td>
<td style="border: 1px solid #bdc3c7; padding: 10px;">✗ Не поливать</td>
<td style="border: 1px solid #bdc3c7; padding: 10px;">✗ Не поливать</td>
<td style="border: 1px solid #bdc3c7; padding: 10px;">✓ Полить</td>
</tr>
</table>

```

```

<div class="code-example">
<table style="border: 3px solid #2c3e50; border-collapse: collapse;">
<tr style="background-color: #34495e; color: white;">
<th style="border: 1px solid #2c3e50; padding: 12px;">Растение</th>
<!-- остальные ячейки -->
</tr>
<!-- остальные строки -->
</table>
</div>
</div>

```

<!-- Пример 4: Творческие рамки -->

```

<div class="example-section">
<h2>4. Творческие варианты рамок</h2>

```

<!-- Градиентная рамка -->

```

<h3>Тень вместо рамки</h3>
<table style="box-shadow: 0 4px 8px rgba(0,0,0,0.1); border-radius: 8px;
overflow: hidden;" class="demo-table">
<tr style="background: linear-gradient(135deg, #667eea, #764ba2); color:
white;">
<th style="padding: 15px;">Увлечение</th>
<th style="padding: 15px;">Время</th>
</tr>

```

```

<tr>
<td style="padding: 12px;">Садоводство</td>
<td style="padding: 12px;">Утро</td>
</tr>
<tr>
<td style="padding: 12px;">Чтение</td>
<td style="padding: 12px;">Вечер</td>
</tr>
</table>


<h3>Рамка с закругленными углами</h3>
<table style="border: 2px solid #e74c3c; border-radius: 15px; overflow: hidden;" class="demo-table">
<tr style="background: #e74c3c; color: white;">
<th style="padding: 12px;">День</th>
<th style="padding: 12px;">Занятие</th>
</tr>
<tr>
<td style="padding: 10px;">Понедельник</td>
<td style="padding: 10px;">HTML</td>
</tr>
<tr>
<td style="padding: 10px;">Среда</td>
<td style="padding: 10px;">CSS</td>
</tr>
</table>
</div>


<div class="example-section">
<h2>🎯 Шпаргалка по рамкам</h2>

<div style="display: grid; grid-template-columns: repeat(auto-fit, minmax(200px, 1fr)); gap: 15px;">
<div style="border: 2px solid #3498db; padding: 10px; text-align: center;">
<strong>Solid</strong><br>сплошная
</div>
<div style="border: 2px dashed #e74c3c; padding: 10px; text-align: center;">
<strong>Dashed</strong><br>пунктирная
</div>
<div style="border: 2px dotted #27ae60; padding: 10px; text-align: center;">
<strong>Dotted</strong><br>точечная
</div>
<div style="border: 4px double #8e44ad; padding: 10px; text-align: center;">

```

```
<strong>Double</strong><br>двойная  
</div>  
</div>  
  
<div class="note">  
<strong>Быстрые примеры:</strong>  
<ul>  
<li><code>border: 1px solid black;</code> - черная сплошная</li>  
<li><code>border: 2px dashed red;</code> - красная пунктирная</li>  
<li><code>border: 3px dotted green;</code> - зеленая точечная</li>  
<li><code>border: 4px double blue;</code> - синяя двойная</li>  
</ul>  
</div>  
</div>  
</body>  
</html>
```

Часть 5: Важные CSS-свойства для рамок

border-collapse — объединение границ

html

```
<table style="border-collapse: collapse;">  
collapse — границы ячеек объединяются (рекомендуется)  
separate — границы ячеек разделены (по умолчанию)
```

Сравнение:

html

```
<!-- С раздельными границами -->  
<table style="border-collapse: separate; border: 2px solid blue;">  
<tr><td style="border: 1px solid red;">Ячейка 1</td></tr>  
</table>
```

```
<!-- С объединенными границами -->  
<table style="border-collapse: collapse; border: 2px solid blue;">  
<tr><td style="border: 1px solid red;">Ячейка 1</td></tr>  
</table>
```

Часть 6: Практическое задание

Задание 1: Создайте таблицу с разными рамками

html

```
<!DOCTYPE html>
```

```
<html>
<head>
<title>Практика с рамками</title>
<style>
.table-exercise {
margin: 20px;
padding: 15px;
border: 1px dashed #ccc;
}
</style>
</head>
<body>
<h1>Практика: Рамки для таблиц</h1>
```

```
<!-- Задание 1: Простая таблица с рамкой -->
<div class="table-exercise">
<h3>1. Таблица с простой черной рамкой</h3>
<table border="1">
<tr>
<th>День</th>
<th>Завтрак</th>
<th>Обед</th>
<th>Ужин</th>
</tr>
<tr>
<td>Понедельник</td>
<td>Каша</td>
<td>Суп</td>
<td>Рыба</td>
</tr>
</table>
</div>
```

```
<!-- Задание 2: Таблица с цветной рамкой -->
<div class="table-exercise">
<h3>2. Таблица с синей рамкой (используйте CSS)</h3>
<table style="border: 2px solid blue;">
<tr>
<th>Книга</th>
<th>Автор</th>
<th>Год</th>
</tr>
<tr>
<td>Война и мир</td>
<td>Лев Толстой</td>
```

```
<td>1869</td>
</tr>
</table>
</div>

<!-- Задание 3: Таблица с пунктирной рамкой --&gt;
&lt;div class="table-exercise"&gt;
&lt;h3&gt;3. Таблица с зеленой пунктирной рамкой&lt;/h3&gt;
<!-- Ваш код здесь --&gt;
&lt;/div&gt;
&lt;/body&gt;
&lt;/html&gt;</pre>
```

Часть 7: Частые ошибки и решения

Ошибка 1: Двойные границы

```
html
<!-- Неправильно -->
<table border="1" style="border: 2px solid red;">
```

Решение: Используйте либо атрибут, либо CSS, но не оба одновременно.

Ошибка 2: Невидимая таблица

```
html
<!-- Рамка есть, но не видна -->
<table style="border: 1px solid white;">
```

Решение: Используйте контрастный цвет.

Ошибка 3: Слишком толстая рамка

```
html
<table style="border: 10px solid black;">
```

Решение: Для большинства таблиц достаточно 1-3px.

Часть 8: Полезные советы

Совет 1: Используйте border-collapse

Всегда добавляйте border-collapse: collapse; для аккуратного вида:

```
html
```

```
<table style="border: 2px solid black; border-collapse: collapse;">
```

Совет 2: Сочетайте с отступами

html

```
<table style="border: 2px solid #333; border-collapse: collapse;">
<tr>
<td style="padding: 10px;">Текст с отступами</td>
</tr>
</table>
```

Совет 3: Тестируйте на разных устройствах

Проверяйте, как выглядит рамка на:

Компьютере

Планшете

Телефоне

Заключение

Что Вы теперь умеете:

- Создавать простые рамки через атрибут border
- Использовать CSS для продвинутых стилей рамок
- Применять разные типы линий (solid, dashed, dotted, double)
- Управлять цветом и толщиной рамок
- Создавать аккуратные таблицы с border-collapse

Ключевые методы создания рамок:

<table border="1"> — самый простой способ
style="border: 2px solid black;" — гибкий CSS-способ

Комбинация с border-collapse: collapse; — для аккуратного вида

Рекомендации по выбору рамки:

-  Для данных — тонкая сплошная рамка
-  Для акцента — цветная или толстая рамка
-  Для форм — пунктирная или точечная рамка
-  Для важных таблиц — двойная рамка

Помните: Хорошо подобранная рамка делает таблицу:

-  Более читаемой



**Визуально привлекательной
Структурированной
Профессиональной**

Следующие шаги:

Попрактикуйтесь с разными стилями рамок
Создайте таблицу с закругленными углами
Поэкспериментируйте с тенями вместо рамок
Добавьте рамки на свои реальные проекты

Теперь Ваши таблицы будут не только функциональными, но и визуально привлекательными! Правильно подобранная рамка может значительно улучшить восприятие информации.

9.4. Практика: Создаём таблицу "Расписание моих занятий" или "Расходы за неделю"

Введение: Превращаем данные в удобные таблицы

Сегодня мы закрепим все знания о таблицах на практике! Мы создадим полезные таблицы, которые Вы сможете использовать в повседневной жизни. Выберите тему, которая Вам ближе — расписание занятий или учёт расходов. Мы разберём обе, шаг за шагом.

Часть 1: Базовая структура таблицы

Начнём с простой таблицы, чтобы вспомнить основные теги.

```
html
<table border="1">
<tr>
<th>День недели</th>
<th>Время</th>
<th>Занятие</th>
</tr>
<tr>
<td>Понедельник</td>
<td>10:00</td>
<td>Йога</td>
</tr>
</table>
```

Результат:

День недели	Время	Занятие
Понедельник	10:00	Йога

Часть 2: Создаём расписание занятий

Давайте сделаем полноценное расписание на неделю с временем и примечаниями.

```
html
<table border="1" style="border-collapse: collapse; width: 100%;">
<tr style="background-color: #4CAF50; color: white;">
```

```

<th style="padding: 12px;">День недели</th>
<th style="padding: 12px;">Время</th>
<th style="padding: 12px;">Занятие</th>
<th style="padding: 12px;">Место</th>
</tr>
<tr>
<td style="padding: 10px;">Понедельник</td>
<td style="padding: 10px;">10:00 - 11:30</td>
<td style="padding: 10px;">Йога для начинающих</td>
<td style="padding: 10px;">Спортзал № 1</td>
</tr>
<tr style="background-color: #f9f9f9;">
<td style="padding: 10px;">Вторник</td>
<td style="padding: 10px;">14:00 - 16:00</td>
<td style="padding: 10px;">Компьютерные курсы</td>
<td style="padding: 10px;">Кабинет 205</td>
</tr>
<tr>
<td style="padding: 10px;">Среда</td>
<td style="padding: 10px;">11:00 - 13:00</td>
<td style="padding: 10px;">Танцы</td>
<td style="padding: 10px;">Актовый зал</td>
</tr>
</table>

```

Результат:

День недели	Время	Занятие	Место
Понедельник	10:00 - 11:30	Йога для начинающих	Спортзал № 1
Вторник	14:00 - 16:00	Компьютерные курсы	Кабинет 205
Среда	11:00 - 13:00	Танцы	Актовый зал

Часть 3: Создаём таблицу расходов за неделю

Теперь давайте сделаем таблицу для контроля бюджета с подсчётом итогов.

html

```

<table border="1" style="border-collapse: collapse; width: 100%;">
<tr style="background-color: #2196F3; color: white;">
<th style="padding: 12px;">День</th>
<th style="padding: 12px;">Категория расходов</th>
<th style="padding: 12px;">Сумма (руб.)</th>
<th style="padding: 12px;">Примечание</th>
</tr>

```

```

<tr>
<td style="padding: 10px;">Понедельник</td>
<td style="padding: 10px;">Продукты</td>
<td style="padding: 10px; text-align: right;">1 850</td>
<td style="padding: 10px;">Молоко, хлеб, овощи</td>
</tr>
<tr style="background-color: #f9f9f9;">
<td style="padding: 10px;">Вторник</td>
<td style="padding: 10px;">Коммунальные услуги</td>
<td style="padding: 10px; text-align: right;">5 240</td>
<td style="padding: 10px;">Квитанция №245</td>
</tr>
<tr>
<td style="padding: 10px;">Среда</td>
<td style="padding: 10px;">Транспорт</td>
<td style="padding: 10px; text-align: right;">600</td>
<td style="padding: 10px;">Проездной на месяц</td>
</tr>
<tr style="background-color: #ffeb3b; font-weight: bold;">
<td style="padding: 10px; colspan="2">ИТОГО за неделю:</td>
<td style="padding: 10px; text-align: right;">7 690</td>
<td style="padding: 10px;">-</td>
</tr>
</table>

```

Результат:

День	Категория расходов	Сумма (руб.)	Примечание
Понедельник	Продукты	1 850	Молоко, хлеб, овощи
Вторник	Коммунальные услуги	5 240	Квитанция №245
Среда	Транспорт	600	Проездной на месяц
ИТОГО за неделю:		7 690	-

Часть 4: Полная страница с примерами

```

html
<!DOCTYPE html>
<html lang="ru">
<head>
<title>Практика: Создаём полезные таблицы</title>
<meta charset="utf-8">
<style>
body {
font-family: 'Arial', sans-serif;
max-width: 1200px;
margin: 0 auto;

```

```
padding: 20px;
line-height: 1.6;
background-color: #f5f5f5;
}
.example-section {
background: white;
padding: 25px;
margin: 25px 0;
border-radius: 10px;
box-shadow: 0 2px 5px rgba(0,0,0,0.1);
}
h1, h2 {
color: #2c3e50;
}
h1 {
border-bottom: 3px solid #3498db;
padding-bottom: 15px;
}
.code-example {
background: #2d2d2d;
color: #f8f8f2;
padding: 15px;
border-radius: 5px;
font-family: 'Courier New', monospace;
margin: 15px 0;
overflow-x: auto;
}
.note {
background: #e7f3ff;
padding: 15px;
border-left: 4px solid #2196F3;
margin: 15px 0;
border-radius: 5px;
}
.demo-table {
margin: 20px 0;
width: 100%;
border-collapse: collapse;
}
</style>
</head>
<body>
<h1> Практика: Создаём полезные таблицы</h1>

<div class="note">
```

💡 Совет: Сегодня мы создадим таблицы, которые Вы сможете использовать в повседневной жизни! Выберите ту, что Вам больше нравится, или создайте обе.

</div>

<!-- Пример 1: Расписание занятий -->

<div class="example-section">
 <h2>🎓 Вариант 1: Расписание моих занятий</h2>

```
<div class="code-example">
<table border="1" style="border-collapse: collapse; width: 100%;">
<tr style="background-color: #4CAF50; color: white;">
<th style="padding: 12px;">День недели</th>
<th style="padding: 12px;">Время</th>
<th style="padding: 12px;">Занятие</th>
<th style="padding: 12px;">Место</th>
</tr>
<tr>
<td style="padding: 10px;">Понедельник</td>
<td style="padding: 10px;">10:00 - 11:30</td>
<td style="padding: 10px;">Йога для начинающих</td>
<td style="padding: 10px;">Спортзал №1</td>
</tr>
<!-- Добавьте свои занятия -->
</table>
</div>
```

```
<table border="1" class="demo-table">
<tr style="background-color: #4CAF50; color: white;">
<th style="padding: 12px;">День недели</th>
<th style="padding: 12px;">Время</th>
<th style="padding: 12px;">Занятие</th>
<th style="padding: 12px;">Место</th>
</tr>
<tr>
<td style="padding: 10px;">Понедельник</td>
<td style="padding: 10px;">10:00 - 11:30</td>
<td style="padding: 10px;">Йога для начинающих</td>
<td style="padding: 10px;">Спортзал №1</td>
</tr>
<tr style="background-color: #f9f9f9;">
<td style="padding: 10px;">Вторник</td>
<td style="padding: 10px;">14:00 - 16:00</td>
<td style="padding: 10px;">Компьютерные курсы</td>
<td style="padding: 10px;">Кабинет 205</td>
```

```

</tr>
<tr>
<td style="padding: 10px;">Среда</td>
<td style="padding: 10px;">11:00 - 13:00</td>
<td style="padding: 10px;">Танцы</td>
<td style="padding: 10px;">Актовый зал</td>
</tr>
<tr style="background-color: #f9f9f9;">
<td style="padding: 10px;">Четверг</td>
<td style="padding: 10px;">15:00 - 16:30</td>
<td style="padding: 10px;">Шахматный клуб</td>
<td style="padding: 10px;">Читальный зал</td>
</tr>
</table>
</div>

```

<!-- Пример 2: Расходы за неделю -->

```

<div class="example-section">
<h2>💰 Вариант 2: Учёт расходов за неделю</h2>

```

```

<div class="code-example">
<table border="1" style="border-collapse: collapse; width: 100%;">
<tr style="background-color: #2196F3; color: white;">
<th style="padding: 12px;">День</th>
<th style="padding: 12px;">Категория расходов</th>
<th style="padding: 12px;">Сумма (руб.)</th>
<th style="padding: 12px;">Примечание</th>
</tr>
<tr>
<td style="padding: 10px;">Понедельник</td>
<td style="padding: 10px;">Продукты</td>
<td style="padding: 10px; text-align: right;">1 850</td>
<td style="padding: 10px;">Молоко, хлеб, овощи</td>
</tr>
</table>
</div>

```

```

<table border="1" class="demo-table">
<tr style="background-color: #2196F3; color: white;">
<th style="padding: 12px;">День</th>
<th style="padding: 12px;">Категория расходов</th>
<th style="padding: 12px;">Сумма (руб.)</th>
<th style="padding: 12px;">Примечание</th>
</tr>
<tr>

```

```

<td style="padding: 10px;">Понедельник</td>
<td style="padding: 10px;">Продукты</td>
<td style="padding: 10px; text-align: right;">1 850</td>
<td style="padding: 10px;">Молоко, хлеб, овощи</td>
</tr>
<tr style="background-color: #f9f9f9;">
<td style="padding: 10px;">Вторник</td>
<td style="padding: 10px;">Коммунальные услуги</td>
<td style="padding: 10px; text-align: right;">5 240</td>
<td style="padding: 10px;">Квитанция №245</td>
</tr>
<tr>
<td style="padding: 10px;">Среда</td>
<td style="padding: 10px;">Транспорт</td>
<td style="padding: 10px; text-align: right;">600</td>
<td style="padding: 10px;">Проездной на месяц</td>
</tr>
<tr style="background-color: #f9f9f9;">
<td style="padding: 10px;">Четверг</td>
<td style="padding: 10px;">Лекарства</td>
<td style="padding: 10px; text-align: right;">450</td>
<td style="padding: 10px;">Аптека №3</td>
</tr>
<tr style="background-color: #ffeb3b; font-weight: bold;">
<td style="padding: 10px; colspan="2">ИТОГО за неделю:</td>
<td style="padding: 10px; text-align: right;">8 140</td>
<td style="padding: 10px;">-</td>
</tr>
</table>
</div>

```

<!-- Часть 5: Практическое задание -->

Практическое задание

 Задание: Создайте свою собственную таблицу. Вы можете использовать шаблоны ниже или придумать свой вариант.

Вариант А: Моё личное расписание

Скопируйте этот код и заполните своими занятиями -->

--	--

```
<th style="padding: 12px;">День недели</th>
<th style="padding: 12px;">Время</th>
<th style="padding: 12px;">Моё занятие</th>
</tr>
<tr>
<td style="padding: 10px;">!-- Ваш день --></td>
<td style="padding: 10px;">!-- Ваше время --></td>
<td style="padding: 10px;">!-- Ваше занятие --></td>
</tr>
</table>
</div>
```

```
<h3>Вариант Б: Мои расходы</h3>
<div class="code-example">
<!-- Заполните таблицу своими расходами --&gt;
&lt;table border="1" style="border-collapse: collapse; width: 100%;"&gt;
&lt;tr style="background-color: #FF9800; color: white;"&gt;
&lt;th style="padding: 12px;"&gt;Дата&lt;/th&gt;
&lt;th style="padding: 12px;"&gt;На что потратил(а)&lt;/th&gt;
&lt;th style="padding: 12px;"&gt;Сумма&lt;/th&gt;
&lt;/tr&gt;
&lt;tr&gt;
&lt;td style="padding: 10px;"&gt;!-- Дата --&gt;&lt;/td&gt;
&lt;td style="padding: 10px;"&gt;!-- Покупка --&gt;&lt;/td&gt;
&lt;td style="padding: 10px;"&gt;!-- Сумма --&gt;&lt;/td&gt;
&lt;/tr&gt;
&lt;/table&gt;
&lt;/div&gt;
&lt;/div&gt;</pre>
```

<!-- Часть 6: Советы по улучшению таблиц --></p>

```
<div class="example-section">
<h2>💡 Советы для лучших таблиц</h2>
```

```
<div style="display: grid; grid-template-columns: repeat(auto-fit, minmax(250px, 1fr)); gap: 20px;">
<div style="border: 2px solid #4CAF50; padding: 15px; border-radius: 8px;">
<h3 style="color: #4CAF50; margin-top: 0;">🎨 Чередуйте строки</h3>
<p>Добавляйте разный фон для четных и нечетных строк — так таблицу легче читать.</p>
<code>style="background-color: #f9f9f9;"</code>
</div>
```

```
<div style="border: 2px solid #2196F3; padding: 15px; border-radius: 8px;">
<h3 style="color: #2196F3; margin-top: 0;">📐 Выравнивайте числа</h3>
```

<p>Числа в столбцах выравнивайте по правому краю — так их проще сравнивать.</p>

```
<code>style="text-align: right;"</code>
</div>
```

<div style="border: 2px solid #FF9800; padding: 15px; border-radius: 8px;">

<h3 style="color: #FF9800; margin-top: 0;">Добавляйте итоги</h3>

<p>Для таблиц с числами всегда добавляйте строку с итоговой суммой.</p>

```
<code>colspan="2"</code>
```

```
</div>
```

```
</div>
```

```
</div>
```

<!-- Часть 7: Шпаргалка по тегам -->

<div class="example-section">

<h2>📋 Шпаргалка по тегам таблиц</h2>

<table border="1" class="demo-table">

<tr style="background-color: #607D8B; color: white;">

<th style="padding: 12px;">Тег</th>

<th style="padding: 12px;">Описание</th>

<th style="padding: 12px;">Пример</th>

```
</tr>
```

```
<tr>
```

<td style="padding: 10px;"><code><table></code></td>

<td style="padding: 10px;">Создает таблицу</td>

<td style="padding: 10px;"><code><table border="1"></code></td>

```
</tr>
```

<tr style="background-color: #f9f9f9;">

<td style="padding: 10px;"><code><tr></code></td>

<td style="padding: 10px;">Создает строку таблицы</td>

<td style="padding: 10px;"><code><tr>...</tr></code></td>

```
</tr>
```

```
<tr>
```

<td style="padding: 10px;"><code><th></code></td>

<td style="padding: 10px;">Заголовок столбца (жирный)</td>

<td style="padding: 10px;"><code><th>День</th></code></td>

```
</tr>
```

<tr style="background-color: #f9f9f9;">

<td style="padding: 10px;"><code><td></code></td>

<td style="padding: 10px;">Обычная ячейка с данными</td>

<td style="padding: 10px;"><code><td>Понедельник</td></code></td>

```
</tr>
```

```
</table>
```

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

Часть 8: Что Вы теперь умеете

- Создавать полезные таблицы для повседневных нужд
- Проектировать структуру таблицы под конкретную задачу
- Добавлять заголовки, данные и примечания
- Использовать чередование цветов для удобства чтения
- Вычислять и показывать итоговые суммы
- Применять colspan для объединения ячеек

Часть 9: Следующие шаги

Для дальнейшего развития:

- Добавьте в таблицу раскрывающиеся подробности
- Создайте таблицу с изображениями (расписание с фото преподавателей)
- Сделайте интерактивную таблицу с возможностью сортировки
- Добавьте подсветку важных строк (например, самые крупные расходы)

Практикуйтесь!

Создайте таблицу для:

- Списка дней рождения родных
- Графика приема лекарств
- Плана ухода за растениями
- Расписания телепередач

Помните: **Лучшая таблица — та, которую Вы действительно используете!** 

Часть 4:

Важные мелочи и публикация сайта

Глава 10: Комментарии в коде — "записки для себя"

10.1. Что такое комментарий <!-- --> и зачем он нужен

Введение: Заметки для себя и других разработчиков

Представьте, что Вы читаете книгу и делаете пометки на полях: "важная мысль", "вернуться к этому позже", "спросить у Марии". Комментарии в HTML — это такие же пометки, но в Вашем коде! Они не видны посетителям сайта, но очень помогают Вам и другим разработчикам разбираться в структуре страницы.

Часть 1: Базовый синтаксис комментариев

Комментарий в HTML создается с помощью специальных символов <!-- и -->. Все, что находится между ними, браузер игнорирует.

html

```
<!-- Эта простой комментарий -->
```

Как это работает:

<!-- — начало комментария

Текст комментария — содержимое, которое не отображается

--> — конец комментария

Часть 2: Практические примеры комментариев

Пример 1: Однострочный комментарий

html

```
<!-- Навигационное меню -->
```

```
<nav>
```

```
<a href="#">Главная</a>
```

```
<a href="#">О нас</a>
```

```
<a href="#">Контакты</a>
```

```
</nav>
```

Пример 2: Многострочный комментарий

html

```
<!--
```

```
Этот раздел содержит контактную информацию
```

Создан: 15.03.2024
Автор: Иванова Мария

-->
<div class="contacts">
<p>Телефон: +7 (123) 456-78-90</p>
</div>

Пример 3: Комментарий в середине строки

html
<h1>Заголовок страницы <!-- Временный заголовок --></h1>

Часть 3: Основные причины использовать комментарии

Давайте рассмотрим, зачем вообще нужны комментарии, на конкретных примерах.

Причина 1: Объяснение сложного кода

html
<!-- Расчет скидки: базовая цена * (1 - процент скидки/100) -->
<div class="price">
Базовая цена: 1000 руб.

Скидка: 15%

Итоговая цена: 850 руб.
</div>

Причина 2: Разметка разделов страницы

html
<!-- ===== HEADER ===== -->
<header>
<!-- Логотип и название сайта -->
<div class="logo">Мой Сайт</div>

<!-- Главное меню навигации -->
<nav class="main-menu">...</nav>
</header>

<!-- ===== MAIN CONTENT ===== -->
<main>
<!-- Блок с приветствием -->
<section class="welcome">...</section>

<!-- Блок с новостями -->

```
<section class="news">...</section>
</main>

<!-- ===== FOOTER ===== -->
<footer>...</footer>
```

Причина 3: Временное отключение кода

```
html
<!--
<nav>
<a href="#">Старое меню</a>
<a href="#">Устаревшая ссылка</a>
</nav>
-->

<nav>
<a href="#">Главная</a>
<a href="#">Новости</a>
<a href="#">Контакты</a>
</nav>
```

Причина 4: Помощь в командной работе

```
html
<!--
РАЗРАБОТЧИК: Петров Алексей
ДАТА: 15.03.2024
ЗАДАЧА: Добавить блок с акциями
СТАТУС: В работе
-->
<section class="promotions">
<!-- TODO: Добавить таймер обратного отсчета -->
<div class="promo-item">...</div>
</section>
```

Часть 4: Полная страница с примерами комментариев

```
html
<!DOCTYPE html>
<html lang="ru">
<head>
<title>Практика: Комментарии в HTML</title>
<meta charset="utf-8">
<style>
```

```
body {
font-family: 'Arial', sans-serif;
max-width: 1000px;
margin: 0 auto;
padding: 20px;
line-height: 1.6;
background-color: #f8f9fa;
}

.example-section {
background: white;
padding: 25px;
margin: 25px 0;
border-radius: 10px;
box-shadow: 0 2px 5px rgba(0,0,0,0.1);
}

h1, h2 {
color: #2c3e50;
}

h1 {
border-bottom: 3px solid #3498db;
padding-bottom: 15px;
}

.code-example {
background: #2d2d2d;
color: #f8f8f2;
padding: 15px;
border-radius: 5px;
font-family: 'Courier New', monospace;
margin: 15px 0;
overflow-x: auto;
}

.note {
background: #e7f3ff;
padding: 15px;
border-left: 4px solid #2196F3;
margin: 15px 0;
border-radius: 5px;
}

.warning {
background: #fff3e0;
padding: 15px;
border-left: 4px solid #ff9800;
margin: 15px 0;
border-radius: 5px;
}
```

```
.comment-demo {  
background: #f0fff0;  
border-left: 4px solid #4CAF50;  
padding: 10px;  
margin: 10px 0;  
}  
</style>  
</head>  
<body>  
<h1>💬 Комментарии в HTML: Ваши заметки в коде</h1>  
  
<div class="note">  
<strong>💡 Основная идея:</strong> Комментарии — это заметки в коде, которые видны только разработчикам. Посетители сайта их не увидят, но они помогут Вам и Вашей команде понимать структуру страницы.  
</div>  
  
<!-- Пример 1: Базовый синтаксис -->  
<div class="example-section">  
<h2>1. Как выглядят комментарии</h2>  
  
<div class="code-example">  
<!-- Это однострочный комментарий -->  
  
<!--  
Это многострочный комментарий.  
Здесь можно писать сколько угодно текста.  
Браузер проигнорирует всё между <!-- и -->  
-->  
  
<div>  
Этот текст увидят пользователи  
<!-- а этот комментарий - только разработчики -->  
</div>  
</div>  
  
<div class="comment-demo">  
<strong>Результат в браузере:</strong><br>  
Этот текст увидят пользователи  
</div>  
</div>  
  
<!-- Пример 2: Практическое использование -->  
<div class="example-section">  
<h2>2. Комментарии в реальном проекте</h2>
```

```
<div class="code-example">
<!DOCTYPE html>
<html lang="ru">
<head>
<meta charset="utf-8">
<title>Мой сайт - Главная</title>
<!-- Подключение стилей -->
<link rel="stylesheet" href="style.css">
</head>
<body>
<!-- ===== ШАПКА САЙТА ===== -->
<header>
<!-- Логотип -->
<div class="logo">МойСайт.ру</div>

<!-- Главное меню -->
<nav class="main-nav">
<a href="#">Главная</a>
<a href="#">О нас</a>
<!-- Временно скрыто
<a href="#">Услуги</a>
-->
<a href="#">Контакты</a>
</nav>
</header>

<!-- ===== ОСНОВНОЕ СОДЕРЖИМОЕ ===== -->
<main>
<!-- Баннер с акцией -->
<section class="promo-banner">
<!-- TODO: Добавить обратный отсчет -->
<h1>Специальное предложение!</h1>
</section>

<!-- Сетка товаров -->
<section class="products">
<!-- ТОВАР 1 -->
<div class="product">
<h3>Название товара 1</h3>
<p>Описание товара</p>
<!-- Цена рассчитывается автоматически -->
<span class="price">1000 руб.</span>
</div>
```

```
<!-- ТОВАР 2 -->
<div class="product">
<h3>Название товара 2</h3>
<p>Описание товара</p>
<span class="price">1500 руб.</span>
</div>
</section>
</main>
```

```
<!-- ===== ПОДВАЛ ===== -->
<footer>
<!-- Контактная информация -->
<div class="contacts">
Телефон: +7 (123) 456-78-90
</div>
<!--
ДОБАВИТЬ:
- Ссылки на соцсети
- Форму обратной связи
- Карту проезда
-->
</footer>
</body>
</html>
</div>
</div>
```

```
<!-- Пример 3: Временное отключение кода -->
<div class="example-section">
<h2>3. Временное отключение элементов</h2>
```

```
<div class="code-example">
<!-- Рабочая навигация -->
<nav>
<a href="#home">Главная</a>
<a href="#about">О нас</a>
<a href="#contact">Контакты</a>
</nav>

<!--
<nav>
<a href="#old-link">Старая страница</a>
<a href="#deprecated">Устаревший раздел</a>
</nav>
-->
```

```
<!--  
<div class="old-banner">  
  <h2>Старый баннер</h2>  
  <p>Эта акция уже завершилась</p>  
</div>  
-->  
</div>  
  
<div class="note">  
  <strong>⚠ Примечание:</strong> Закомментированный код не  
удаляется, а просто временно отключается. Это удобно для  
тестирования или если Вы хотите сохранить старую версию на случай  
отката.  
</div>  
</div>  
  
<!-- Пример 4: Отладочные комментарии -->  
<div class="example-section">  
  <h2>4. Комментарии для отладки</h2>  
  
  <div class="code-example">  
    <div class="user-profile">  
      <!-- ОТЛАДКА: Проверка данных пользователя -->  
      <!--  
        Имя: Иван Петров  
        Email: ivan@example.com  
        Регистрация: 15.03.2023  
      -->  
  
      <h2>Профиль пользователя</h2>  
      <div class="user-info">  
        <!-- FIXME: Добавить аватар пользователя -->  
        <div class="avatar">Фото</div>  
  
        <!-- TODO: Сделать проверку заполненности полей -->  
        <div class="name">Иван Петров</div>  
        <div class="email">ivan@example.com</div>  
      </div>  
  
      <!-- BUG: При обновлении профиля теряются настройки -->  
      <button class="save-btn">Сохранить изменения</button>  
    </div>  
  </div>
```

```
<div style="display: grid; grid-template-columns: repeat(auto-fit, minmax(200px, 1fr)); gap: 15px; margin: 15px 0;">
<div style="border: 2px solid #4CAF50; padding: 10px; border-radius: 5px;">
<strong style="color: #4CAF50;">TODO</strong><br>Запланированная задача
</div>
<div style="border: 2px solid #FF9800; padding: 10px; border-radius: 5px;">
<strong style="color: #FF9800;">FIXME</strong><br>Требуется исправление
</div>
<div style="border: 2px solid #F44336; padding: 10px; border-radius: 5px;">
<strong style="color: #F44336;">BUG</strong><br>Обнаруженная ошибка
</div>
<div style="border: 2px solid #2196F3; padding: 10px; border-radius: 5px;">
<strong style="color: #2196F3;">NOTE</strong><br>Важное примечание
</div>
</div>
</div>
```

<!-- Пример 5: Частые ошибки -->

```
<div class="example-section">
<h2>5. Что нельзя делать с комментариями</h2>
```

```
<div class="warning">
<strong>⚠ Внимание!</strong> Эти примеры показывают неправильное использование комментариев.
</div>
```

Ошибка 1: Незакрытый комментарий

```
<div class="code-example">
<!-- Этот комментарий никогда не заканчивается
<div>
Весь этот код не будет работать!
Потому что комментарий не закрыт
</div>
</div>
```

Ошибка 2: Комментарии внутри тегов

```
<div class="code-example">
<div <!-- этот комментарий сломает код --> class="header">
Содержимое
</div>
</div>
```

Ошибка 3: Вложенные комментарии

```
<div class="code-example">
<!--
Внешний комментарий
<!-- Внутренний комментарий -->
Текст между комментариями
-->
</div>

<div class="note">
<strong>✓ Правильно:</strong> Комментарии должны быть правильно
закрыты и не могут быть вложенными.
</div>
</div>

<!-- Практическое задание -->
<div class="example-section">
<h2>🎯 Практическое задание</h2>

<div class="code-example">
<!DOCTYPE html>
<html lang="ru">
<head>
<meta charset="utf-8">
<title>Моё расписание</title>
</head>
<body>
<h1>Моё ежедневное расписание</h1>

<!-- Ваша ЗАДАЧА: Добавить комментарии ниже -->

<div class="morning">
<h2>Утро</h2>
<ul>
<li>07:00 - Подъем</li>
<li>07:30 - Завтрак</li>
<li>08:30 - Прогулка</li>
</ul>
</div>

<div class="day">
<h2>День</h2>
<ul>
<li>12:00 - Обед</li>
<li>13:00 - Компьютерные курсы</li>
<li>15:00 - Чтение</li>
```

```

</ul>
</div>

<div class="evening">
<h2>Вечер</h2>
<ul>
<li>18:00 - Ужин</li>
<li>19:00 - Телевизор</li>
<li>21:00 - Подготовка ко сну</li>
</ul>
</div>
</body>
</html>
</div>

<div class="note">
<strong>📝 Задание:</strong> Добавьте в код выше комментарии, которые:
<ul>
<li>Объясняют структуру страницы</li>
<li>Отмечают наиболее важные дела</li>
<li>Показывают, какие пункты временные</li>
<li>Добавляют заметки для будущих улучшений</li>
</ul>
</div>
</div>

<!-- Шпаргалка -->
<div class="example-section">
<h2>📋 Шпаргалка по комментариям</h2>

<table border="1" style="border-collapse: collapse; width: 100%; margin: 15px 0;">
<tr style="background-color: #34495e; color: white;">
<th style="padding: 12px;">Синтаксис</th>
<th style="padding: 12px;">Назначение</th>
<th style="padding: 12px;">Пример</th>
</tr>
<tr>
<td style="padding: 10px;"><code><!-- текст --></code></td>
<td style="padding: 10px;">Обычный комментарий</td>
<td style="padding: 10px;"><code><!-- меню --></code></td>
</tr>
<tr style="background-color: #f9f9f9;">

```

```
<td style="padding: 10px;"><code><!-- <br>многострочный <br>--></code></td>
<td style="padding: 10px;">Многострочный комментарий</td>
<td style="padding: 10px;"><code><!-- раздел<br>новостей --></code></td>
</tr>
<tr>
<td style="padding: 10px;"><code><!-- TODO --></code></td>
<td style="padding: 10px;">Запланированная задача</td>
<td style="padding: 10px;"><code><!-- TODO: добавить фото --></code></td>
</tr>
<tr style="background-color: #f9f9f9;">
<td style="padding: 10px;"><code><!-- FIXME --></code></td>
<td style="padding: 10px;">Требуется исправление</td>
<td style="padding: 10px;"><code><!-- FIXME: исправить баг --></code></td>
</tr>
</table>
</div>
</body>
</html>
```

Часть 6: Что Вы теперь умеете

-  Понимать назначение HTML-комментариев
-  Создавать однострочные и многострочные комментарии
-  Использовать комментарии для разметки разделов страницы
-  Временно отключать код с помощью комментариев
-  Добавлять напоминания и задачи (TODO, FIXME)
-  Избегать распространенных ошибок с комментариями

Часть 7: Итоги и рекомендации

Ключевые моменты:

Комментарии начинаются с <!-- и заканчиваются -->

Они не видны пользователям на странице

Помогают организовать код и работать в команде

Позволяют временно отключать элементы

Профессиональные советы:

Комментируйте структуру — разделяйте комментариями основные блоки страницы

Объясняйте сложные моменты — если код делает что-то неочевидное, добавьте комментарий

Используйте стандартные пометки — TODO для задач, FIXME для исправлений

Не переусердствуйте — хороший код часто понятен и без комментариев

Следите за актуальностью — удаляйте устаревшие комментарии



Следующие шаги:

Практикуйтесь добавлять комментарии в свои проекты

Используйте разные типы комментариев (разделители, пояснения, задачи)

Научитесь быстро комментировать/раскомментировать код в редакторе

Изучите комментарии в CSS и JavaScript

Помните: Хорошо прокомментированный код — признак заботливого разработчика! 

10.2. Как комментарии помогают не запутаться в своём коде

Введение: Спасательный круг в море кода

Представьте, что Вы открываете старый шкаф с вещами. Если всё аккуратно разложено по полочкам с подписями "летняя одежда", "зимние вещи", "документы" — Вы быстро найдёте нужное. Если же всё свалкой — придётся перерывать каждый угол. Комментарии в коде — это такие же "подписи на полочках", которые помогают быстро ориентироваться и не запутаться.

Часть 1: Проблема — код без комментариев

Давайте посмотрим на реальный пример, как легко запутаться в коде без комментариев.

```
html
<!DOCTYPE html>
<html>
<head>
<title>Сайт</title>
<style>
.b1 { background: #f0f0f0; padding: 20px; margin: 10px; }
.b2 { border: 1px solid #ccc; border-radius: 5px; }
.b3 { display: flex; justify-content: space-between; }
.c1 { color: #333; font-size: 18px; }
.c2 { color: #666; font-size: 14px; }
</style>
</head>
<body>
<div class="b1 b2">
<div class="b3">
<div class="c1">Текст</div>
<div class="c2">Другой текст</div>
</div>
</div>
<div class="b1">
<div>Ещё текст</div>
</div>
</body>
</html>
```

Что здесь происходит? Непонятно! Классы с непонятными названиями, непонятная структура. Теперь представим, что этот код написан 3 месяца назад — Вы вообще ничего не вспомните!

Часть 2: Решение — код с комментариями

А теперь тот же код, но с комментариями:

```
html
<!DOCTYPE html>
<html>
<head>
<title>Сайт компании - Главная</title>
<style>
/* === СТИЛИ ДЛЯ КАРТОЧЕК === */
.card-container {
background: #f0f0f0;
padding: 20px;
margin: 10px;
}
.card-border {
border: 1px solid #ccc;
border-radius: 5px;
}

/* === СТИЛИ ДЛЯ ШАПКИ КАРТОЧКИ === */
.card-header {
display: flex;
justify-content: space-between;
}

/* === ТИПОГРАФИКА === */
.card-title {
color: #333;
font-size: 18px;
}
.card-subtitle {
color: #666;
font-size: 14px;
}
</style>
</head>
<body>
<!-- КАРТОЧКА ПРОДУКТА --&gt;
&lt;div class="card-container card-border"&gt;
<!-- Заголовок карточки с названием и ценой --&gt;
&lt;div class="card-header"&gt;
&lt;div class="card-title"&gt;iPhone 15&lt;/div&gt;</pre>
```

```
<div class="card-subtitle">85 000 руб.</div>
</div>
</div>

<!-- КАРТОЧКА АКЦИИ -->
<div class="card-container">
<div>Скидка 10% на все смартфоны</div>
</div>
</body>
</html>
```

Теперь всё понятно! Комментарии превратили хаос в порядок.

Часть 3: Конкретные способы как комментарии предотвращают путаницу

Способ 1: Разметка крупных блоков

```
html
<!-- ===== HEADER ===== -->
<header>
<!-- ЛОГОТИП И НАЗВАНИЕ -->
<div class="logo">

</div>

<!-- ПОИСК ПО САЙТУ -->
<div class="search">
<input type="text" placeholder="Поиск товаров...">
<button>Найти</button>
</div>

<!-- КОРЗИНА ПОКУПОК -->
<div class="cart">
<span>Корзина (3)</span>
</div>
</header>

<!-- ===== MAIN CONTENT ===== -->
>
<main>
<!-- БАННЕР АКЦИИ -->
<section class="promo-banner">
<h1>Распродажа до 50%</h1>
</section>
```

```

<!-- КАТАЛОГ ТОВАРОВ -->
<section class="catalog">
<!-- ФИЛЬТРЫ -->
<aside class="filters">
<!-- фильтры по цене, бренду и т.д. -->
</aside>

<!-- СПИСОК ТОВАРОВ -->
<div class="products-grid">
<!-- товары -->
</div>
</section>
</main>

<!-- ===== FOOTER ===== -->
<footer>
<!-- КОНТАКТЫ -->
<div class="contacts">...</div>

<!-- НАВИГАЦИЯ -->
<nav class="footer-nav">...</nav>
</footer>

```

Способ 2: Объяснение сложной логики

```

html
<!-- КАЛЬКУЛЯТОР СКИДКИ -->
<div class="discount-calculator">
<!--
ФОРМУЛА РАСЧЕТА:
Итоговая цена = Базовая цена × (1 - Скидка/100)
Если сумма покупки > 50000, добавляется доп. скидка 5%
-->
<div class="price-inputs">
<label>Базовая цена:</label>
<input type="number" id="basePrice" value="1000">

<label>Скидка %:</label>
<input type="number" id="discount" value="10">
</div>

<!-- РЕЗУЛЬТАТ РАСЧЕТА -->
<div class="result">
<!-- TODO: Добавить отображение этапов расчета -->

```

Итоговая цена: 900 руб.

```
<!--  
ДОПОЛНИТЕЛЬНАЯ ЛОГИКА:  
- Если цена > 1000 - показываем "Выгодно!"  
- Если скидка > 50% - выделяем красным  
-->  
<div id="priceNote" class="note">Выгодно!</div>  
</div>  
</div>
```

Способ 3: Отслеживание изменений и версий

html

```
<!--  
ИСТОРИЯ ИЗМЕНЕНИЙ:  
2024-03-15 - Иванов - Добавил блок акций  
2024-03-20 - Петрова - Изменил цвета акции  
2024-03-25 - Сидоров - Добавил таймер обратного отсчета
```

СЛЕДУЮЩИЕ ЗАДАЧИ:

- Добавить слайдер акций
- Сделать адаптивную верстку для мобильных
- Интегрировать с системой скидок

-->

```
<section class="promotions">  
<!-- АКЦИЯ НЕДЕЛИ -->  
<div class="weekly-promo">  
<h2>Акция недели</h2>  
<!-- FIXME: Таймер иногда сбрасывается при обновлении страницы -->  
<div class="timer" id="promoTimer">05:00:00</div>  
</div>
```

```
<!-- СТАРАЯ АКЦИЯ (ВРЕМЕННО ОТКЛЮЧЕНО)  
<div class="old-promo">  
<h2>Прошлая акция</h2>  
<p>Уже завершилась</p>  
</div>  
-->  
</section>
```

Часть 4: Полная страница с примерами организации кода

html

```
<!DOCTYPE html>
```

```
<html lang="ru">
<head>
<title>Как не запутаться в коде: Практика комментариев</title>
<meta charset="utf-8">
<style>
body {
font-family: 'Arial', sans-serif;
max-width: 1200px;
margin: 0 auto;
padding: 20px;
line-height: 1.6;
background-color: #f5f5f5;
}
.example-section {
background: white;
padding: 25px;
margin: 25px 0;
border-radius: 10px;
box-shadow: 0 2px 5px rgba(0,0,0,0.1);
}
h1, h2 {
color: #2c3e50;
}
h1 {
border-bottom: 3px solid #3498db;
padding-bottom: 15px;
}
.code-example {
background: #2d2d2d;
color: #f8f8f2;
padding: 15px;
border-radius: 5px;
font-family: 'Courier New', monospace;
margin: 15px 0;
overflow-x: auto;
}
.note {
background: #e7f3ff;
padding: 15px;
border-left: 4px solid #2196F3;
margin: 15px 0;
border-radius: 5px;
}
.warning {
background: #ffea7;
```

```
padding: 15px;
border-left: 4px solid #fdcb6e;
margin: 15px 0;
border-radius: 5px;
}
.success {
background: #d1f7d1;
padding: 15px;
border-left: 4px solid #4CAF50;
margin: 15px 0;
border-radius: 5px;
}

```

</style>

</head>

<body>

<h1> Комментарии как навигация: Как не заблудиться в своём коде</h1>

<div class="note">

💡 Основная идея: Комментарии — это карта Вашего кода. Без них легко заблудиться, особенно когда Вы возвращаетесь к проекту через несколько месяцев или работаете в команде.

</div>

<!-- Пример 1: До и после -->

<div class="example-section">

<h2>1. Превращение хаоса в порядок</h2>

<div class="warning">

✖ БЕЗ КОММЕНТАРИЕВ — ХАОС:

</div>

<div class="code-example">

<div class="b1">

<div class="c3">

Заголовок

<div class="b2">

<div class="i1"></div>

<div class="i2"></div>

</div>

</div>

<div class="c4">

<button class="btn1">Кнопка</button>

</div>

</div>

</div>

```
<div class="success">
<strong>✓ С КОММЕНТАРИЯМИ — ПОРЯДОК:</strong>
</div>
<div class="code-example">
<!-- КАРТОЧКА ТОВАРА -->
<div class="product-card">

<!-- ЗАГОЛОВОК КАРТОЧКИ -->
<div class="card-header">
<span class="product-title">Заголовок</span>

<!-- РЕЙТИНГ И ИКОНКИ -->
<div class="product-meta">
<div class="rating-stars"></div>
<div class="action-icons"></div>
</div>
</div>

<!-- КНОПКА ДОБАВЛЕНИЯ В КОРЗИНУ -->
<div class="card-actions">
<button class="add-to-cart-btn">Кнопка</button>
</div>
</div>
</div>
</div>

<!-- Пример 2: Реальная структура сайта -->
<div class="example-section">
<h2>2. Организация крупного проекта</h2>

<div class="code-example">
<!DOCTYPE html>
<html lang="ru">
<head>
<meta charset="utf-8">
<title>Интернет-магазин электроники</title>

<!--
СТРУКТУРА СТРАНИЦЫ:
1. Шапка с лого, поиском и корзиной
2. Основной контент: баннер, фильтры, товары
3. Боковая панель: рекомендации
4. Подвал: контакты и навигация
-->
```

```
</head>
<body>
<!-- ===== ШАПКА САЙТА ===== -->
<header class="main-header">
<!-- ЛОГОТИП И НАЗВАНИЕ -->
<div class="logo-section">

<h1>TechStore</h1>
</div>

<!-- ПОИСКОВАЯ СИСТЕМА -->
<div class="search-section">
<!--
ПОИСК РАБОТАЕТ ПО:
- Названию товара
- Категории
- Бренду
TODO: Добавить поиск по характеристикам
-->
<input type="text" placeholder="Найти товар...">
<button>Поиск</button>
</div>

<!-- КОРЗИНА И ПОЛЬЗОВАТЕЛЬ -->
<div class="user-section">
<div class="cart">Корзина (3)</div>
<div class="user-profile">Профиль</div>
</div>
</header>

<!-- ===== ОСНОВНОЙ КОНТЕНТ ===== -->
<main class="main-content">

<!-- БАННЕР ГЛАВНОЙ АКЦИИ -->
<section class="hero-banner">
<!-- FIXME: Сделать баннер кликабельным -->
<h2>Распродажа смартфонов до 40%</h2>
<p>Только до конца недели</p>
</section>

<!-- КАТАЛОГ ТОВАРОВ -->
<section class="product-catalog">

<!-- ПАНЕЛЬ ФИЛЬТРОВ -->
<aside class="filter-sidebar">
```

```
<!--  
ФИЛЬТРЫ ПО:  
- Цене (диапазон)  
- Бренду (чекбоксы)  
- Рейтингу (звезды)  
- Наличию (в наличии/под заказ)  
-->  
<div class="price-filter">  
<h3>Цена</h3>  
<input type="range" min="0" max="100000">  
</div>  
  
<div class="brand-filter">  
<h3>Бренд</h3>  
<!-- Список брендов -->  
</div>  
</aside>  
  
<!-- СЕТКА ТОВАРОВ -->  
<div class="products-grid">  
<!--  
КАЖДАЯ КАРТОЧКА ТОВАРА СОДЕРЖИТ:  
- Изображение  
- Название  
- Цена  
- Рейтинг  
- Кнопка "В корзину"  
-->  
<div class="product-card">  
<!-- ТОВАР 1 -->  
  
<h3>iPhone 15</h3>  
<span class="price">85 000 руб.</span>  
<div class="rating">★★★★☆</div>  
<button>В корзину</button>  
</div>  
  
<!-- ТОВАР 2 -->  
<div class="product-card">  
  
<h3>Samsung Galaxy S24</h3>  
<span class="price">65 000 руб.</span>  
<div class="rating">★★★★★</div>  
<button>В корзину</button>  
</div>
```

```
<!--  
TODO: Добавить пагинацию  
FIXME: Исправить выравнивание карточек  
-->  
</div>  
</section>  
</main>  
  
<!-- ===== ПОДВАЛ САЙТА ===== -->  
<footer class="main-footer">  
<!--  
РАЗРАБОТЧИК: Иванов А.С.  
ВЕРСИЯ: 1.2.3  
ДАТА ПОСЛЕДНЕГО ОБНОВЛЕНИЯ: 2024-03-15  
-->  
<div class="footer-content">  
<div class="contacts">  
<h3>Контакты</h3>  
<p>Телефон: +7 (123) 456-78-90</p>  
</div>  
  
<nav class="footer-nav">  
<h3>Навигация</h3>  
<a href="#">О компании</a>  
<a href="#">Доставка</a>  
<a href="#">Гарантия</a>  
</nav>  
</div>  
</footer>  
</body>  
</html>  
</div>  
</div>  
  
<!-- Пример 3: Система пометок -->  
<div class="example-section">  
<h2>3. Система пометок для самоорганизации</h2>  
  
<div style="display: grid; grid-template-columns: repeat(auto-fit, minmax(250px, 1fr)); gap: 15px; margin: 20px 0;">  
<div style="border: 2px solid #4CAF50; padding: 15px; border-radius: 8px;">  
<h3 style="color: #4CAF50; margin-top: 0;">✓ TODO</h3>  
<p><strong>Задачи на будущее</strong></p>  
<code><!-- TODO: Добавить слайдер --></code>
```

```
<code><!-- TODO: Сделать адаптив --></code>
</div>

<div style="border: 2px solid #FF9800; padding: 15px; border-radius: 8px;">
<h3 style="color: #FF9800; margin-top: 0;">⚠ FIXME</h3>
<p><strong>Требует исправления</strong></p>
<code><!-- FIXME: Исправить баг с кнопкой --></code>
<code><!-- FIXME: Устранить утечку памяти --></code>
</div>

<div style="border: 2px solid #F44336; padding: 15px; border-radius: 8px;">
<h3 style="color: #F44336; margin-top: 0;">🐞 BUG</h3>
<p><strong>Обнаруженные ошибки</strong></p>
<code><!-- BUG: На iOS не работает --></code>
<code><!-- BUG: Падает при быстром клике --></code>
</div>

<div style="border: 2px solid #2196F3; padding: 15px; border-radius: 8px;">
<h3 style="color: #2196F3; margin-top: 0;">💡 NOTE</h3>
<p><strong>Важные заметки</strong></p>
<code><!-- NOTE: Кэшируется 24 часа --></code>
<code><!-- NOTE: Требуются права админа --></code>
</div>
</div>

<div class="code-example">
<div class="user-profile">
<!-- TODO: Добавить загрузку аватара -->
<!-- FIXME: При обновлении теряются настройки -->
<!-- BUG: На мобильных съезжает layout -->
<!-- NOTE: Данные кэшируются 1 час -->

<div class="avatar">
<!-- TODO: Сделать preview перед загрузкой -->

</div>

<div class="user-info">
<!-- FIXME: Проверить валидацию email -->
<input type="email" placeholder="Email">

<!-- NOTE: Пароль должен быть от 8 символов -->
<input type="password" placeholder="Пароль">
</div>
</div>
```

```
</div>
</div>

<!-- Пример 4: Временное отключение кода --&gt;
&lt;div class="example-section"&gt;
&lt;h2&gt;4. Управление версиями и экспериментами&lt;/h2&gt;

&lt;div class="code-example"&gt;
&lt;div class="payment-system"&gt;
&lt;!--
ИСТОРИЯ ИЗМЕНЕНИЙ:
2024-01-15 - Добавлена оплата картой
2024-02-20 - Добавлен PayPal
2024-03-10 - Временно отключен Google Pay (баг)
2024-03-15 - Добавлена оплата через СБП
--&gt;

&lt;h3&gt;Способы оплаты&lt;/h3&gt;

&lt;!-- АКТИВНЫЕ СПОСОБЫ ОПЛАТЫ --&gt;
&lt;div class="active-methods"&gt;
&lt;!-- БАНКОВСКАЯ КАРТА --&gt;
&lt;div class="payment-method"&gt;
&lt;img src="visa-mc.png" alt="Карты"&gt;
&lt;span&gt;Банковская карта&lt;/span&gt;
&lt;/div&gt;

&lt;!-- PAYPAL --&gt;
&lt;div class="payment-method"&gt;
&lt;img src="paypal.png" alt="PayPal"&gt;
&lt;span&gt;PayPal&lt;/span&gt;
&lt;/div&gt;

&lt;!-- СБП --&gt;
&lt;div class="payment-method"&gt;
&lt;img src="sbp.png" alt="СБП"&gt;
&lt;span&gt;СБП&lt;/span&gt;
&lt;/div&gt;
&lt;/div&gt;

&lt;!-- ВРЕМЕННО ОТКЛЮЧЕННЫЕ МЕТОДЫ --&gt;
&lt;!--
&lt;div class="disabled-methods"&gt;
&lt;!-- GOOGLE PAY (ОТКЛЮЧЕН ИЗ-ЗА БАГА) --&gt;
&lt;div class="payment-method"&gt;</pre>
```

```

<span>Google Pay</span>
</div>

<!-- APPLE PAY (В РАЗРАБОТКЕ) -->
<div class="payment-method">

<span>Apple Pay</span>
</div>
</div>
-->

<!-- ЭКСПЕРИМЕНТАЛЬНЫЕ ФИЧИ -->
<!--
<div class="experimental">
<!-- КРИПТОВАЛЮТЫ (ТЕСТИРУЕТСЯ) -->
<div class="payment-method">

<span>Криптовалюты</span>
</div>
</div>
-->
</div>
</div>
</div>

<!-- Практическое задание -->
<div class="example-section">
<h2>🎯 Практическое задание: Наведите порядок в коде</h2>

<div class="code-example">
<!DOCTYPE html>
<html>
<head>
<title>Сайт</title>
<style>
.s1 { max-width: 1200px; margin: 0 auto; padding: 20px; }
.h1 { background: #2c3e50; color: white; padding: 20px; }
.c1 { display: flex; justify-content: space-between; }
.n1 { display: flex; gap: 20px; }
.m1 { display: grid; grid-template-columns: 1fr 3fr; gap: 30px; margin: 40px 0; }
.f1 { background: #34495e; color: white; padding: 30px; margin-top: 50px; }
</style>
</head>
```

```
<body>
<div class="s1">
<div class="h1">
<div class="c1">
<div>Лого</div>
<div class="n1">
<div>Главная</div>
<div>О нас</div>
<div>Услуги</div>
<div>Контакты</div>
</div>
</div>
</div>

<div class="m1">
<div>
<h3>Категории</h3>
<ul>
<li>Категория 1</li>
<li>Категория 2</li>
<li>Категория 3</li>
</ul>
</div>
<div>
<h2>Заголовок</h2>
<p>Текст страницы...</p>
</div>
</div>

<div class="f1">
<div>Телефон: +7 (123) 456-78-90</div>
<div>Email: info@site.ru</div>
</div>
</div>
</body>
</html>
</div>

<div class="note">
<strong>📝 Задание:</strong> Добавьте комментарии в код выше, чтобы:
<ul>
<li>Разметить основные разделы страницы</li>
<li>Объяснить назначение каждого блока</li>
<li>Добавить TODO для улучшений</li>
<li>Создать систему пометок для будущих изменений</li>
```

```
</ul>
</div>
</div>

<!-- Шпаргалка -->


## Шпаргалка по организации кода



| Ситуация             | Как комментировать    | Пример                                            |
|----------------------|-----------------------|---------------------------------------------------|
| Разделы страницы     | Разделители с ===     | <code>&lt;!-- === HEADER === --&gt;</code>        |
| Сложная логика       | Подробное объяснение  | <code>&lt;!-- Расчет: цена × скидка --&gt;</code> |
| Будущие задачи       | TODO пометки          | <code>&lt;!-- TODO: Добавить фото --&gt;</code>   |
| Проблемы             | FIXME/BUG пометки     | <code>&lt;!-- FIXME: Исправить баг --&gt;</code>  |
| Временное отключение | Закомментировать блок | <code>&lt;!-- старый код --&gt;</code>            |


```

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

Часть 5: Что Вы теперь умеете

- Превращать хаотичный код в структурированный
- Создавать навигацию по коду с помощью комментариев
- Использовать систему пометок (TODO, FIXME, BUG)
- Отслеживать историю изменений в коде
- Временно отключать код без его удаления
- Объяснять сложную логику для себя и других

Часть 6: Итоги и рекомендации

Ключевые принципы организации кода:

Комментируйте разделы — используйте визуальные разделители

Объясняйте сложное — если логика неочевидна, добавьте комментарий

Отмечайте задачи — используйте TODO для планирования

Фиксируйте проблемы — FIXME и BUG для отслеживания ошибок

Ведите историю — комментарии с датами и авторами изменений

Профессиональные привычки:

Комментируйте по мере написания — не откладывайте "на потом"

Обновляйте комментарии — устаревшие комментарии хуже их отсутствия

Используйте единый стиль — договоритесь о правилах в команде

Не комментируйте очевидное — <!-- кнопка --> у кнопки излишне

Результат правильного комментирования:

Возвращаться к коду через месяцы и сразу понимать его

Работать в команде без лишних вопросов "а что это?"

Быстро находить и исправлять проблемы

Экономить время на понимании собственного кода

Помните: **Хорошо организованный код с комментариями** — это Ваш личный гид, который никогда не даст Вам заблудиться! 

10.3. Практика: добавляем комментарии в свой код, чтобы объяснить, что делает каждый раздел

Введение: Превращаем код в понятную инструкцию

Сегодня мы возьмём реальный HTML-код и превратим его из "чёрного ящика" в хорошо документированную инструкцию. Вы научитесь комментировать каждый раздел так, чтобы даже новичок мог понять структуру и назначение всех элементов страницы.

Часть 1: Исходный код без комментариев

Давайте начнём с типичной страницы, которая есть у многих пенсионеров — страницы с расписанием и полезными ссылками.

```
html
<!DOCTYPE html>
<html>
<head>
<title>Мой ежедневник</title>
<style>
body { font-family: Arial; max-width: 800px; margin: 0 auto; padding: 20px; }
.header { background: #2c3e50; color: white; padding: 20px; text-align: center; }
.nav { background: #34495e; padding: 15px; }
.nav a { color: white; margin: 0 15px; text-decoration: none; }
.main { display: grid; grid-template-columns: 1fr 2fr; gap: 20px; margin: 20px 0; }
.sidebar { background: #ecf0f1; padding: 20px; border-radius: 5px; }
.content { padding: 20px; }
.schedule { border-collapse: collapse; width: 100%; }
.schedule th, .schedule td { border: 1px solid #ddd; padding: 12px; text-align: left; }
.schedule th { background: #3498db; color: white; }
.footer { background: #7f8c8d; color: white; text-align: center; padding: 20px; margin-top: 30px; }
.links li { margin: 10px 0; }
</style>
</head>
<body>
<div class="header">
<h1>Мой ежедневник</h1>
<p>Органайзер для пенсионера</p>
</div>
```

```
<div class="nav">
<a href="#schedule">Расписание</a>
<a href="#links">Полезные ссылки</a>
<a href="#contacts">Контакты</a>
</div>

<div class="main">
<div class="sidebar">
<h3>Ближайшие события</h3>
<ul>
<li>Посещение врача - 15:00</li>
<li>Встреча с друзьями - 18:00</li>
<li>Приём лекарств - 20:00</li>
</ul>

<h3>Быстрые заметки</h3>
<textarea placeholder="Запишите важное..." rows="4" style="width: 100%;"></textarea>
</div>

<div class="content">
<h2>Моё расписание на неделю</h2>
<table class="schedule">
<tr>
<th>День</th>
<th>Время</th>
<th>Занятие</th>
<th>Место</th>
</tr>
<tr>
<td>Понедельник</td>
<td>10:00-11:30</td>
<td>Йога</td>
<td>Спортзал</td>
</tr>
<tr>
<td>Вторник</td>
<td>14:00-16:00</td>
<td>Компьютерные курсы</td>
<td>Библиотека</td>
</tr>
<tr>
<td>Среда</td>
<td>11:00-12:30</td>
<td>Танцы</td>
```

```
<td>Клуб</td>
</tr>
</table>

<h2>Полезные ссылки</h2>
<ul class="links">
<li><a href="#">Госуслуги</a></li>
<li><a href="#">Социальные службы</a></li>
<li><a href="#">Поликлиника</a></li>
<li><a href="#">Аптека</a></li>
</ul>
</div>
</div>

<div class="footer">
<p>Телефон: +7 (123) 456-78-90 | Email: pensioner@example.com</p>
<p>© 2024 Мой ежедневник. Все права защищены.</p>
</div>
</body>
</html>
```

Проблема: Код работает, но через месяц Вы забудете, зачем создавали каждый блок и как он работает!

Часть 2: Пошаговое комментирование кода

Теперь давайте превратим этот код в хорошо документированную страницу.

```
html
<!DOCTYPE html>
<html>
<head>
<!--
ЗАГОЛОВОК СТРАНИЦЫ И МЕТА-ИНФОРМАЦИЯ
Здесь указывается то, что не видно пользователю, но важно для
браузера
-->
<title>Мой ежедневник - Органайзер для пенсионера</title>
<meta charset="utf-8">

<!--
СТИЛИ СТРАНИЦЫ
Описываем внешний вид всех элементов
TODO: Вынести стили в отдельный файл для удобства
-->
```

```
-->
<style>
/* ОСНОВНЫЕ НАСТРОЙКИ СТРАНИЦЫ */
body {
font-family: Arial, sans-serif; /* Шрифт для лучшей читаемости */
max-width: 800px;           /* Ограничение ширины для удобства чтения */
*/
margin: 0 auto;             /* Центрирование страницы */
padding: 20px;              /* Отступы от краёв */
background-color: #f9f9f9;   /* Светлый фон для уменьшения нагрузки
на глаза */
}

/* ШАПКА САЙТА */
.header {
background: #2c3e50;        /* Тёмно-синий цвет для контраста */
color: white;
padding: 20px;
text-align: center;
border-radius: 5px 5px 0 0;  /* Закругленные углы сверху */
}

/* НАВИГАЦИОННОЕ МЕНЮ */
.nav {
background: #34495e;        /* Чуть светлее шапки */
padding: 15px;
text-align: center;
}
.nav a {
color: white;
margin: 0 15px;
text-decoration: none;      /* Убираем подчёркивание у ссылок */
font-weight: bold;
}
.nav a:hover {
text-decoration: underline; /* Подчёркиваем при наведении */
}

/* ОСНОВНАЯ ОБЛАСТЬ КОНТЕНТА */
.main {
display: grid;              /* Используем CSS Grid для разметки */
grid-template-columns: 1fr 2fr; /* Боковая панель:контент = 1:2 */
gap: 20px;                  /* Расстояние между колонками */
margin: 20px 0;
}
```

```
/* БОКОВАЯ ПАНЕЛЬ */
.sidebar {
background: #ecf0f1;           /* Светло-серый фон */
padding: 20px;
border-radius: 5px;
box-shadow: 0 2px 5px rgba(0,0,0,0.1); /* Лёгкая тень для объема */
}

/* ОСНОВНОЕ СОДЕРЖИМОЕ */
.content {
padding: 20px;
background: white;            /* Белый фон для контента */
border-radius: 5px;
box-shadow: 0 2px 5px rgba(0,0,0,0.1);
}

/* ТАБЛИЦА РАСПИСАНИЯ */
.schedule {
border-collapse: collapse;    /* Убираем двойные границы */
width: 100%;
}
.schedule th, .schedule td {
border: 1px solid #ddd;        /* Светло-серые границы */
padding: 12px;
text-align: left;
}
.schedule th {
background: #3498db;          /* Синий фон для заголовков */
color: white;
}
.schedule tr:nth-child(even) {
background: #f2f2f2;           /* Чередование цвета строк для удобства
чтения */
}

/* ПОДВАЛ САЙТА */
.footer {
background: #7f8c8d;           /* Серый цвет */
color: white;
text-align: center;
padding: 20px;
margin-top: 30px;
border-radius: 0 0 5px 5px;    /* Закругленные углы снизу */
}
```

```
/* СПИСОК ССЫЛОК */
.links li {
margin: 10px 0;
}
.links a {
color: #2980b9; /* Синий цвет ссылок */
text-decoration: none;
}
.links a:hover {
text-decoration: underline;
}
</style>
</head>
<body>
<!-- ===== ШАПКА САЙТА ===== -->
<div class="header">
<h1>Мой ежедневник</h1>
<p>Органайзер для пенсионера</p>
<!--
ЗДЕСЬ МОЖНО ДОБАВИТЬ:
- Фотографию пользователя
- Текущую дату и время
- Погоду
-->
</div>

<!-- ===== ГЛАВНОЕ МЕНЮ ===== -->
<div class="nav">
<!--
НАВИГАЦИЯ ПО РАЗДЕЛАМ САЙТА
Ссылки ведут к якорям на странице (#section)
-->
<a href="#schedule">Расписание</a>
<a href="#links">Полезные ссылки</a>
<a href="#contacts">Контакты</a>
<!-- TODO: Добавить выпадающее меню для мобильных устройств -->
</div>

<!-- ===== ОСНОВНОЙ КОНТЕНТ
===== -->
<div class="main">
<!--
БОКОВАЯ ПАНЕЛЬ - ДЛЯ БЫСТРОГО ДОСТУПА
-->
```

Содержит часто используемую информацию

-->

```
<div class="sidebar">
<!-- БЛИЖАЙШИЕ СОБЫТИЯ НА СЕГОДНЯ --&gt;
&lt;h3&gt;Ближайшие события&lt;/h3&gt;
&lt;ul&gt;
&lt;li&gt;Посещение врача - 15:00&lt;/li&gt;
&lt;li&gt;Встреча с друзьями - 18:00&lt;/li&gt;
&lt;li&gt;Приём лекарств - 20:00&lt;/li&gt;
&lt;!--</pre>
```

В БУДУЩЕМ МОЖНО:

- Добавить автоматическое определение текущих событий
- Синхронизировать с календарём

-->

```
</ul>
```

```
<!-- БЫСТРЫЕ ЗАМЕТКИ -->
```

```
<h3>Быстрые заметки</h3>
```

<!--

ТЕКСТОВОЕ ПОЛЕ ДЛЯ ЭКСПРЕСС-ЗАМЕТОК

Пользователь может быстро записать важную информацию

FIXME: Добавить сохранение заметок при обновлении страницы

-->

```
<textarea placeholder="Запишите важное..." rows="4" style="width: 100%;"></textarea>
```

```
</div>
```

<!--

ОСНОВНАЯ ОБЛАСТЬ СОДЕРЖИМОГО

Здесь размещается основная информация страницы

-->

```
<div class="content">
```

```
<!-- РАЗДЕЛ РАСПИСАНИЯ -->
```

```
<h2 id="schedule">Моё расписание на неделю</h2>
```

<!--

ТАБЛИЦА РАСПИСАНИЯ

Отображает регулярные занятия пользователя

Структура: День → Время → Занятие → Место

-->

```
<table class="schedule">
```

```
<tr>
```

```
<th>День</th>
```

```
<th>Время</th>
```

```
<th>Занятие</th>
```

```
<th>Место</th>
```

```
</tr>

<tr>
<td>Понедельник</td>
<td>10:00-11:30</td>
<td>Йога</td>
<td>Спортзал</td>
</tr>

<tr>
<td>Вторник</td>
<td>14:00-16:00</td>
<td>Компьютерные курсы</td>
<td>Библиотека</td>
</tr>

<tr>
<td>Среда</td>
<td>11:00-12:30</td>
<td>Танцы</td>
<td>Клуб</td>
</tr>
<!--
TODO: Добавить строки для остальных дней недели
FIXME: Сделать возможность редактирования расписания
-->
</table>
```

```
<!-- РАЗДЕЛ ПОЛЕЗНЫХ ССЫЛОК -->
<h2 id="links">Полезные ссылки</h2>
<!--
КОЛЛЕКЦИЯ ЧАСТО ИСПОЛЬЗУЕМЫХ ССЫЛОК
Пользователь может быстро перейти к важным ресурсам
-->
<ul class="links">
<li><a href="https://gosuslugi.ru" target="_blank">Госуслуги</a></li>
<li><a href="https://szn.gov.ru" target="_blank">Социальные
службы</a></li>
<li><a href="#">Поликлиника № 123</a></li>
<li><a href="#">Аптека "Здоровье"</a></li>
<!--
NOTE: target="_blank" открывает ссылку в новой вкладке
TODO: Добавить иконки для ссылок
TODO: Сделать возможность добавления новых ссылок
-->
```

```
</ul>
</div>
</div>

<!-- ===== ПОДВАЛ САЙТА ===== -->
<div class="footer" id="contacts">
<!--
КОНТАКТНАЯ ИНФОРМАЦИЯ И КОПИРАЙТ
Размещается внизу каждой страницы
-->
<p>Телефон: +7 (123) 456-78-90 | Email: pensioner@example.com</p>
<p>© 2024 Мой ежедневник. Все права защищены.</p>
<!--
ДОПОЛНИТЕЛЬНАЯ ИНФОРМАЦИЯ ДЛЯ ПОДВАЛА:
- Ссылки на социальные сети
- Форма обратной связи
- Карта проезда
-->
</div>

<!--
===== СКРИПТЫ И ДОПОЛНИТЕЛЬНЫЙ КОД
=====

ЗДЕСЬ МОГУТ РАЗМЕЩАТЬСЯ:
- JavaScript код для интерактивности
- Аналитические счётчики
- Вспомогательные функции

TODO: Добавить JavaScript для:
1. Сохранения заметок в локальное хранилище
2. Динамического обновления текущих событий
3. Редактирования расписания

ПРИМЕЧАНИЕ: В данный момент JavaScript не используется для
простоты
-->
</body>
</html>
```

Часть 3: Полная страница с практическими заданиями

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

```
<html lang="ru">
<head>
<title>Практика: Комментирование HTML-кода</title>
<meta charset="utf-8">
<style>
body {
font-family: 'Arial', sans-serif;
max-width: 1200px;
margin: 0 auto;
padding: 20px;
line-height: 1.6;
background-color: #f5f5f5;
}
.example-section {
background: white;
padding: 25px;
margin: 25px 0;
border-radius: 10px;
box-shadow: 0 2px 5px rgba(0,0,0,0.1);
}
h1, h2 {
color: #2c3e50;
}
h1 {
border-bottom: 3px solid #3498db;
padding-bottom: 15px;
}
.code-example {
background: #2d2d2d;
color: #f8f8f2;
padding: 15px;
border-radius: 5px;
font-family: 'Courier New', monospace;
margin: 15px 0;
overflow-x: auto;
}
.note {
background: #e7f3ff;
padding: 15px;
border-left: 4px solid #2196F3;
margin: 15px 0;
border-radius: 5px;
}
.practice-task {
background: #fff3e0;
```

```
padding: 20px;  
border-left: 4px solid #ff9800;  
margin: 20px 0;  
border-radius: 5px;  
}  
.comment-example {  
background: #f0fff0;  
padding: 10px;  
margin: 10px 0;  
border-left: 4px solid #4CAF50;  
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h1>📝 Практика: Добавляем комментарии в свой код</h1>
```

```
<div class="note">
```

```
<strong>🎯 Цель урока:</strong> Научиться комментировать каждый раздел кода так, чтобы через полгода Вы могли легко понять, что делает каждая часть страницы и зачем она нужна.
```

```
</div>
```

```
<!-- Пример готового комментирования -->
```

```
<div class="example-section">
```

```
<h2>1. Пример правильно закомментированного кода</h2>
```

```
<div class="code-example">
```

```
<!DOCTYPE html>
```

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

```
<head>
```

```
<!--
```

```
МЕТА-ИНФОРМАЦИЯ СТРАНИЦЫ
```

```
Здесь указывается служебная информация для браузера и поисковых систем
```

```
-->
```

```
<meta charset="utf-8">
```

```
<meta name="viewport" content="width=device-width, initial-scale=1.0">
```

```
<title>Мой садовый дневник</title>
```

```
<!--
```

```
ПОДКЛЮЧЕНИЕ СТИЛЕЙ
```

```
TODO: Разделить стили на отдельные файлы для разных разделов
```

```
-->
```

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

```
</head>
```

```
<body>
<!-- ===== ШАПКА ===== -->
<header class="main-header">
<!-- ЛОГОТИП И НАЗВАНИЕ САЙТА -->
<div class="logo">
<h1>Мой садовый дневник</h1>
<p>Учёт растений и работ в саду</p>
</div>

<!-- НАВИГАЦИЯ ПО РАЗДЕЛАМ -->
<nav class="main-nav">
<a href="#plants">Мои растения</a>
<a href="#schedule">График работ</a>
<a href="#notes">Заметки</a>
</nav>
</header>

<!-- ===== ОСНОВНОЕ СОДЕРЖИМОЕ
===== -->
<main class="main-content">

<!-- РАЗДЕЛ: СПИСОК РАСТЕНИЙ -->
<section id="plants" class="plants-section">
<h2>Мои растения</h2>

<!-- КАРТОЧКИ РАСТЕНИЙ -->
<div class="plants-grid">
<!-- КАРТОЧКА ТОМАТОВ -->
<div class="plant-card">
<h3>Помидоры</h3>
<!--
ИНФОРМАЦИЯ ОБ УХОДЕ:
- Дата посадки
- График полива
- Особенности ухода
-->
<p><strong>Посажены:</strong> 15 мая 2024</p>
<p><strong>Полив:</strong> Каждый день</p>
<!-- TODO: Добавить фото растения -->
</div>

<!-- КАРТОЧКА ОГУРЦОВ -->
<div class="plant-card">
<h3>Огурцы</h3>
<p><strong>Посажены:</strong> 20 мая 2024</p>
```

```
<p><strong>Полив:</strong> Через день</p>
</div>
</div>
</section>
</main>

<!-- ===== ПОДВАЛ ===== -->
<footer class="main-footer">
<!--
КОНТАКТНАЯ ИНФОРМАЦИЯ:
- Телефон для экстренной связи
- Email для обратной связи
- Адрес участка
-->
<div class="contact-info">
<p>Телефон: +7 (123) 456-78-90</p>
<p>Участок: Садовое товарищество "Ромашка"</p>
</div>
</footer>
```

```
<!--
===== СКРИПТЫ =====
```

ЗДЕСЬ БУДУТ ПОДКЛЮЧАТЬСЯ JavaScript ФАЙЛЫ:
- main.js - основные функции
- plants.js - работа с растениями
- schedule.js - управление расписанием

NOTE: В данный момент скрипты не подключены для простоты
-->

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

```
<!-- Практическое задание 1 -->
<div class="practice-task">
<h2> Практическое задание 1: Комментируем таблицу расходов</h2>

<div class="code-example">
<!DOCTYPE html>
<html>
<head>
<title>Учёт расходов</title>
<style>
```

```
body { font-family: Arial; margin: 20px; }
.table { width: 100%; border-collapse: collapse; }
.table th, .table td { border: 1px solid #ddd; padding: 8px; }
.table th { background: #4CAF50; color: white; }
.total { font-weight: bold; background: #f2f2f2; }
.income { color: green; }
.expense { color: red; }
</style>
</head>
<body>
<h1>Учёт моих расходов</h1>

<table class="table">
<tr>
<th>Дата</th>
<th>Категория</th>
<th>Описание</th>
<th>Сумма</th>
</tr>
<tr>
<td>01.03.2024</td>
<td>Продукты</td>
<td>Молоко, хлеб, овощи</td>
<td class="expense">850 руб.</td>
</tr>
<tr>
<td>01.03.2024</td>
<td>Пенсия</td>
<td>Ежемесячная пенсия</td>
<td class="income">25 000 руб.</td>
</tr>
<tr class="total">
<td colspan="3">ИТОГО:</td>
<td>24 150 руб.</td>
</tr>
</table>

<div>
<h3>Добавить новую запись</h3>
<form>
<input type="date">
<select>
<option>Продукты</option>
<option>Коммунальные услуги</option>
<option>Транспорт</option>
```

```
<option>Пенсия</option>
</select>
<input type="text" placeholder="Описание">
<input type="number" placeholder="Сумма">
<button>Добавить</button>
</form>
</div>
</body>
</html>
</div>
```

```
<div class="note">
<strong>📝 Задание:</strong> Добавьте комментарии к этому коду, объясняющие:
<ul>
<li>Назначение каждого раздела страницы</li>
<li>Структуру таблицы расходов</li>
<li>Разные цвета для доходов и расходов</li>
<li>Назначение формы добавления записей</li>
<li>Планы по улучшению (TODO)</li>
</ul>
</div>
</div>
```

```
<!-- Практическое задание 2 -->
<div class="practice-task">
<h2>🎯 Практическое задание 2: Комментируем страницу с рецептами</h2>
```

```
<div class="code-example">
<!DOCTYPE html>
<html>
<head>
<title>Мои рецепты</title>
<style>
.container { max-width: 800px; margin: 0 auto; }
.recipe { border: 1px solid #ddd; padding: 20px; margin: 20px 0; }
.ingredients { background: #f9f9f9; padding: 15px; }
.steps { margin: 15px 0; }
.step { margin: 10px 0; }
.time { color: #666; font-style: italic; }
.difficulty { display: inline-block; padding: 5px 10px; border-radius: 15px; }
.easy { background: #d4edda; color: #155724; }
.medium { background: #fff3cd; color: #856404; }
</style>
```

```
</head>
<body>
<div class="container">
<h1>Моя кулинарная книга</h1>

<div class="recipe">
<h2>Борщ</h2>
<div class="time">Время приготовления: 2 часа</div>
<span class="difficulty easy">Простой</span>

<div class="ingredients">
<h3>Ингредиенты:</h3>
<ul>
<li>Свекла - 2 шт</li>
<li>Картофель - 3 шт</li>
<li>Капуста - 200г</li>
<li>Мясо - 400г</li>
</ul>
</div>

<div class="steps">
<h3>Приготовление:</h3>
<div class="step">1. Порезать овощи</div>
<div class="step">2. Сварить мясо</div>
<div class="step">3. Добавить овощи</div>
<div class="step">4. Варить 1.5 часа</div>
</div>
</div>

<div class="recipe">
<h2>Оладьи</h2>
<div class="time">Время приготовления: 30 минут</div>
<span class="difficulty medium">Средний</span>

<div class="ingredients">
<h3>Ингредиенты:</h3>
<ul>
<li>Мука - 2 стакана</li>
<li>Яйца - 2 шт</li>
<li>Молоко - 1 стакан</li>
<li>Сахар - 2 ст. ложки</li>
</ul>
</div>
</div>
</div>
```

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

<div class="note">
<strong>📝 Задание:</strong> Добавьте комментарии, которые объяснят:
<ul>
<li>Структуру карточки рецепта</li>
<li>Систему обозначения сложности приготовления</li>
<li>Разделение на ингредиенты и шаги приготовления</li>
<li>Идеи для будущих улучшений</li>
</ul>
</div>
</div>

<!-- Шпаргалка по комментированию --&gt;
&lt;div class="example-section"&gt;
&lt;h2&gt;📝 Шпаргалка: Что комментировать в коде&lt;/h2&gt;

&lt;table border="1" style="border-collapse: collapse; width: 100%; margin: 15px 0;"&gt;
&lt;tr style="background-color: #34495e; color: white;"&gt;
&lt;th style="padding: 12px;"&gt;Что комментировать&lt;/th&gt;
&lt;th style="padding: 12px;"&gt;Пример комментария&lt;/th&gt;
&lt;th style="padding: 12px;"&gt;Зачем нужно&lt;/th&gt;
&lt;/tr&gt;
&lt;tr&gt;
&lt;td style="padding: 10px;"&gt;Крупные разделы&lt;/td&gt;
&lt;td style="padding: 10px;"&gt;&lt;code&gt;&lt;!-- === HEADER === --&gt;&lt;/code&gt;&lt;/td&gt;
&lt;td style="padding: 10px;"&gt;Быстрая навигация по коду&lt;/td&gt;
&lt;/tr&gt;
&lt;tr style="background-color: #f9f9f9;"&gt;
&lt;td style="padding: 10px;"&gt;Сложная логика&lt;/td&gt;
&lt;td style="padding: 10px;"&gt;&lt;code&gt;&lt;!-- Расчет: цена × кол-во --&gt;&lt;/code&gt;&lt;/td&gt;
&lt;td style="padding: 10px;"&gt;Объяснение не очевидных моментов&lt;/td&gt;
&lt;/tr&gt;
&lt;tr&gt;
&lt;td style="padding: 10px;"&gt;Назначение классов&lt;/td&gt;
&lt;td style="padding: 10px;"&gt;&lt;code&gt;&lt;!-- Основной контейнер --&gt;&lt;/code&gt;&lt;/td&gt;
&lt;td style="padding: 10px;"&gt;Понимание структуры стилей&lt;/td&gt;
&lt;/tr&gt;
&lt;tr style="background-color: #f9f9f9;"&gt;
&lt;td style="padding: 10px;"&gt;Будущие улучшения&lt;/td&gt;</pre>
```

```
<td style="padding: 10px;"><code><!-- TODO: Добавить фото --></code></td>
<td style="padding: 10px;">Планирование развития</td>
</tr>
<tr>
<td style="padding: 10px;">Проблемные места</td>
<td style="padding: 10px;"><code><!-- FIXME: Исправить баг --></code></td>
<td style="padding: 10px;">Отслеживание ошибок</td>
</tr>
</table>
```

```
<div class="comment-example">
<h3> Пример хорошего комментария:</h3>
<code><!--
```

РАЗДЕЛ: ГАЛЕРЕЯ ФОТОГРАФИЙ

Отображает фотографии в сетке 3 колонки

При нажатии на фото открывается полноразмерная версия

TODO: Добавить ленивую загрузку для оптимизации

```
--></code>
```

```
</div>
```

```
</div>
```

```
</body>
```

```
</html>
```

Часть 4: Что Вы теперь умеете

-  **Комментировать структуру HTML-страницы**
-  **Объяснять назначение каждого раздела**
-  **Добавлять TODO-заметки для будущих улучшений**
-  **Отмечать проблемные места (FIXME, BUG)**
-  **Создавать навигацию по коду с помощью комментариев**
-  **Делать код понятным для себя и других**

Часть 5: Итоги и рекомендации

Ключевые принципы комментирования:

Комментируйте разделы — используйте визуальные разделители для крупных блоков

Объясняйте сложное — если логика не очевидна, добавьте пояснение

Отмечайте задачи — используйте TODO для планирования улучшений

Фиксируйте проблемы — FIXME и BUG для отслеживания ошибок

Описывайте назначение — зачем нужен этот блок, что он делает

Практические советы:

Комментируйте по ходу написания — не откладывайте "на потом"

Используйте единый стиль — договоритесь с собой о правилах

Обновляйте комментарии — устаревшие комментарии вводят в заблуждение

Не бойтесь подробностей — лучше больше объяснений, чем меньше



Результат правильного комментирования:

Через полгода вы поймёте свой код с первого взгляда

Другие люди смогут разобраться в Вашем коде

Легче вносить изменения — понимаете последствия

Проще находить ошибки — комментарии подсказывают, что должно работать

Помните: Хорошо закомментированный код — это Ваша личная инструкция, которая всегда подскажет, что Вы делали и зачем!



Глава 11:

Формы — для общения с посетителями

11.1. Простая форма <form>

Введение: Ваш первый диалог с сайтом

Представьте, что Вы заполняете анкету в поликлинике, пишете заявление или оставляете запись в гостевой книге. Формы в HTML — это именно то, что позволяет пользователям "разговаривать" с сайтом: отправлять данные, задавать вопросы, оставлять отзывы. Сегодня мы научимся создавать такие формы!

Часть 1: Базовая структура формы

Любая форма начинается с тега `<form>` и заканчивается `</form>`. Всё, что находится между этими тегами, является частью формы.

```
html
<form>
<!-- Здесь будут поля для ввода --&gt;
&lt;/form&gt;</pre>
```

Пока что это просто "конверт" для будущих данных. Чтобы форма стала полезной, добавим основные атрибуты:

```
html
<form action="/обработать-данные" method="POST">
<!-- Поля формы --&gt;
&lt;/form&gt;</pre>
```

Разберём атрибуты:

`action` — куда отправлять данные (адрес сервера)
`method` — как отправлять данные (GET или POST)

Часть 2: Основные элементы формы

2.1. Текстовое поле (<input type="text">)

```
html
<form>
<!-- Простое текстовое поле --&gt;
&lt;label&gt;Ваше имя:&lt;/label&gt;
&lt;input type="text" name="username" placeholder="Введите ваше имя"&gt;</pre>
```

```
<!-- Поле для email -->
<label>Ваш email:</label>
<input type="email" name="useremail" placeholder="example@mail.ru">
</form>
```

Результат:

text
Ваше имя: [Введите ваше имя]
Ваш email: [example@mail.ru]

2.2. Поле пароля (<input type="password">)

html
<form>
<label>Пароль:</label>
<input type="password" name="password" placeholder="Придумайте пароль">
</form>

Результат:

text
Пароль: [.....]

2.3. Кнопка отправки (<input type="submit">)

html
<form>
<!-- Поля формы -->
<input type="submit" value="Отправить данные">
</form>

Часть 3: Практический пример — форма регистрации

Давайте создадим полноценную форму регистрации на сайте.

html
<form action="/register" method="POST">
<h3>Регистрация на сайте</h3>

<!-- Текстовое поле для имени -->
<label>Имя и фамилия:</label>
<input type="text" name="fullname" placeholder="Иван Иванов" required>

```

<!-- Поле для email -->
<label>Email адрес:</label>
<input type="email" name="email" placeholder="ivan@example.com" required>

<!-- Поле для пароля -->
<label>Пароль:</label>
<input type="password" name="password" placeholder="Не менее 6 символов" required>

<!-- Выпадающий список -->
<label>Ваш город:</label>
<select name="city">
<option value="">Выберите город</option>
<option value="moscow">Москва</option>
<option value="spb">Санкт-Петербург</option>
<option value="ekb">Екатеринбург</option>
</select>

<!-- Радиокнопки -->
<label>Пол:</label>
<input type="radio" name="gender" value="male" id="male">
<label for="male">Мужской</label>

<input type="radio" name="gender" value="female" id="female">
<label for="female">Женский</label>

<!-- Чекбокс -->
<input type="checkbox" name="newsletter" id="news">
<label for="news">Подписаться на рассылку</label>

<!-- Кнопка отправки -->
<input type="submit" value="Зарегистрироваться">
</form>

```

Часть 4: Полная страница с различными формами

```

html
<!DOCTYPE html>
<html lang="ru">
<head>
<title>Практика: Создание HTML-форм</title>
<meta charset="utf-8">
<style>
body {

```

```
font-family: 'Arial', sans-serif;
max-width: 1000px;
margin: 0 auto;
padding: 20px;
line-height: 1.6;
background-color: #f8f9fa;
}
.example-section {
background: white;
padding: 25px;
margin: 25px 0;
border-radius: 10px;
box-shadow: 0 2px 5px rgba(0,0,0,0.1);
}
h1, h2 {
color: #2c3e50;
}
h1 {
border-bottom: 3px solid #3498db;
padding-bottom: 15px;
}
.code-example {
background: #2d2d2d;
color: #f8f8f2;
padding: 15px;
border-radius: 5px;
font-family: 'Courier New', monospace;
margin: 15px 0;
overflow-x: auto;
}
.note {
background: #e7f3ff;
padding: 15px;
border-left: 4px solid #2196F3;
margin: 15px 0;
border-radius: 5px;
}
.warning {
background: #fff3e0;
padding: 15px;
border-left: 4px solid #ff9800;
margin: 15px 0;
border-radius: 5px;
}
```

```
/* Стили для демонстрационных форм */
.demo-form {
background: #f8f9fa;
padding: 20px;
border-radius: 8px;
margin: 20px 0;
}
.form-group {
margin-bottom: 15px;
}
label {
display: block;
margin-bottom: 5px;
font-weight: bold;
color: #2c3e50;
}
input[type="text"],
input[type="email"],
input[type="password"],
input[type="tel"],
select,
textarea {
width: 100%;
padding: 10px;
border: 2px solid #ddd;
border-radius: 5px;
font-size: 16px;
box-sizing: border-box;
}
input[type="text"]:focus,
input[type="email"]:focus,
input[type="password"]:focus,
select:focus,
textarea:focus {
border-color: #3498db;
outline: none;
}
input[type="submit"],
input[type="reset"],
button {
background: #3498db;
color: white;
padding: 12px 25px;
border: none;
border-radius: 5px;
```

```
cursor: pointer;
font-size: 16px;
margin-right: 10px;
}
input[type="submit"]:hover,
button:hover {
background: #2980b9;
}
.radio-group, .checkbox-group {
margin: 10px 0;
}
.radio-group label, .checkbox-group label {
display: inline-block;
margin-right: 15px;
font-weight: normal;
}
</style>
</head>
<body>
<h1> Формы в HTML: Начинаем общение с сайтом</h1>
```

```
<div class="note">
<strong> Основная идея:</strong> Формы — это мост между
пользователем и сайтом. С их помощью люди могут регистрироваться,
оставлять комментарии, делать заказы и выполнять множество других
действий.
</div>
```

<!-- Пример 1: Базовая структура формы -->

```
<div class="example-section">
<h2>1. Базовая структура формы</h2>
```

```
<div class="code-example">
<!-- Простейшая форма -->
<form action="/обработчик" method="POST">
<!-- Поля для ввода данных -->
<label for="name">Ваше имя:</label>
<input type="text" id="name" name="username">

<!-- Кнопка отправки -->
<input type="submit" value="Отправить">
</form>
</div>
```

```
<div class="demo-form">
```

```
<h3>Демонстрация:</h3>
<form style="background: white; padding: 15px; border-radius: 5px;">
<label for="demo-name">Ваше имя:</label>
<input type="text" id="demo-name" name="username" placeholder="Введите ваше имя" style="width: 100%; padding: 8px; margin-bottom: 10px;">

<input type="submit" value="Отправить" style="background: #3498db; color: white; padding: 10px 20px; border: none; border-radius: 3px; cursor: pointer;">
</form>
</div>

<div class="note">
<strong>📋 Объяснение атрибутов:</strong>
<ul>
<li><code>action</code> - куда отправлять данные (URL сервера)</li>
<li><code>method</code> - способ отправки (GET или POST)</li>
<li><code>name</code> - имя поля (используется на сервере)</li>
<li><code>id</code> - уникальный идентификатор (для связи с label)</li>
</ul>
</div>
</div>

<!-- Пример 2: Различные типы полей --&gt;
&lt;div class="example-section"&gt;
&lt;h2&gt;2. Различные типы полей ввода&lt;/h2&gt;

&lt;div class="code-example"&gt;
&lt;form action="/submit" method="POST"&gt;
<!-- Текстовое поле --&gt;
&lt;label&gt;Обычный текст:&lt;/label&gt;
&lt;input type="text" name="text_field" placeholder="Введите текст"&gt;

<!-- Поле для email --&gt;
&lt;label&gt;Email:&lt;/label&gt;
&lt;input type="email" name="email_field" placeholder="example@mail.com"&gt;

<!-- Поле для пароля --&gt;
&lt;label&gt;Пароль:&lt;/label&gt;
&lt;input type="password" name="password_field"&gt;

<!-- Числовое поле --&gt;
&lt;label&gt;Возраст:&lt;/label&gt;
&lt;input type="number" name="age" min="0" max="120"&gt;</pre>
```

```
<!-- Телефон -->
<label>Телефон:</label>
<input type="tel" name="phone" pattern="[0-9]{10}">
</form>
</div>

<div class="demo-form">
<h3>Демонстрация разных полей:</h3>
<form style="background: white; padding: 15px; border-radius: 5px;">
<div class="form-group">
<label>Обычный текст:</label>
<input type="text" placeholder="Введите текст" style="width: 100%; padding: 8px;">
</div>

<div class="form-group">
<label>Email:</label>
<input type="email" placeholder="example@mail.com" style="width: 100%; padding: 8px;">
</div>

<div class="form-group">
<label>Пароль:</label>
<input type="password" placeholder="....." style="width: 100%; padding: 8px;">
</div>

<div class="form-group">
<label>Возраст:</label>
<input type="number" min="0" max="120" style="width: 100%; padding: 8px;">
</div>

<div class="form-group">
<label>Телефон:</label>
<input type="tel" placeholder="+7 (123) 456-78-90" style="width: 100%; padding: 8px;">
</div>
</form>
</div>
</div>

<!-- Пример 3: Практическая форма регистрации -->
<div class="example-section">
<h2>3. Практический пример: Форма регистрации</h2>
```

```
<div class="code-example">
<form action="/register" method="POST">
<h3>Регистрация нового пользователя</h3>

<!-- Группа полей для личных данных --&gt;
&lt;fieldset&gt;
&lt;legend&gt;Личные данные&lt;/legend&gt;

&lt;div class="form-group"&gt;
&lt;label for="fullname"&gt;ФИО:&lt;/label&gt;
&lt;input type="text" id="fullname" name="fullname" required&gt;
&lt;/div&gt;

&lt;div class="form-group"&gt;
&lt;label for="birthdate"&gt;Дата рождения:&lt;/label&gt;
&lt;input type="date" id="birthdate" name="birthdate"&gt;
&lt;/div&gt;
&lt;/fieldset&gt;

<!-- Группа полей для контактов --&gt;
&lt;fieldset&gt;
&lt;legend&gt;Контактная информация&lt;/legend&gt;

&lt;div class="form-group"&gt;
&lt;label for="email"&gt;Email:&lt;/label&gt;
&lt;input type="email" id="email" name="email" required&gt;
&lt;/div&gt;

&lt;div class="form-group"&gt;
&lt;label for="phone"&gt;Телефон:&lt;/label&gt;
&lt;input type="tel" id="phone" name="phone"&gt;
&lt;/div&gt;
&lt;/fieldset&gt;

<!-- Кнопки --&gt;
&lt;input type="submit" value="Зарегистрироваться"&gt;
&lt;input type="reset" value="Очистить форму"&gt;
&lt;/form&gt;
&lt;/div&gt;

&lt;div class="demo-form"&gt;
&lt;h3&gt;Демонстрация формы регистрации:&lt;/h3&gt;
&lt;form style="background: white; padding: 20px; border-radius: 5px;"&gt;</pre>
```

```
<fieldset style="border: 1px solid #ddd; padding: 15px; margin-bottom: 15px; border-radius: 5px;">
<legend style="padding: 0 10px; font-weight: bold;">Личные данные</legend>
```

```
<div class="form-group">
<label for="demo-fullname">ФИО:</label>
<input type="text" id="demo-fullname" name="fullname" placeholder="Иванов Иван Иванович" required style="width: 100%; padding: 8px;">
</div>
```

```
<div class="form-group">
<label for="demo-birthdate">Дата рождения:</label>
<input type="date" id="demo-birthdate" name="birthdate" style="width: 100%; padding: 8px;">
</div>
</fieldset>
```

```
<fieldset style="border: 1px solid #ddd; padding: 15px; margin-bottom: 15px; border-radius: 5px;">
<legend style="padding: 0 10px; font-weight: bold;">Контактная информация</legend>
```

```
<div class="form-group">
<label for="demo-email">Email:</label>
<input type="email" id="demo-email" name="email" placeholder="ivanov@example.com" required style="width: 100%; padding: 8px;">
</div>
```

```
<div class="form-group">
<label for="demo-phone">Телефон:</label>
<input type="tel" id="demo-phone" name="phone" placeholder="+7 (123) 456-78-90" style="width: 100%; padding: 8px;">
</div>
</fieldset>
```

```
<input type="submit" value="Зарегистрироваться" style="background: #27ae60; color: white; padding: 10px 20px; border: none; border-radius: 3px; cursor: pointer; margin-right: 10px;">
<input type="reset" value="Очистить форму" style="background: #e74c3c; color: white; padding: 10px 20px; border: none; border-radius: 3px; cursor: pointer;">
</form>
```

```
</div>
</div>

<!-- Пример 4: Форма обратной связи --&gt;
&lt;div class="example-section"&gt;
&lt;h2&gt;4. Форма обратной связи&lt;/h2&gt;

&lt;div class="code-example"&gt;
&lt;form action="/feedback" method="POST"&gt;
&lt;h3&gt;Обратная связь&lt;/h3&gt;

&lt;div class="form-group"&gt;
&lt;label for="subject"&gt;Тема сообщения:&lt;/label&gt;
&lt;select id="subject" name="subject" required&gt;
&lt;option value=""&gt;Выберите тему&lt;/option&gt;
&lt;option value="question"&gt;Вопрос&lt;/option&gt;
&lt;option value="complaint"&gt;Жалоба&lt;/option&gt;
&lt;option value="suggestion"&gt;Предложение&lt;/option&gt;
&lt;/select&gt;
&lt;/div&gt;

&lt;div class="form-group"&gt;
&lt;label for="message"&gt;Ваше сообщение:&lt;/label&gt;
&lt;textarea id="message" name="message" rows="5"
placeholder="Опишите Вашу проблему или вопрос..." required&gt;&lt;/textarea&gt;
&lt;/div&gt;

&lt;div class="form-group"&gt;
&lt;label&gt;Срочность:&lt;/label&gt;
&lt;div class="radio-group"&gt;
&lt;input type="radio" id="low" name="urgency" value="low"&gt;
&lt;label for="low"&gt;Низкая&lt;/label&gt;

&lt;input type="radio" id="medium" name="urgency" value="medium" checked&gt;
&lt;label for="medium"&gt;Средняя&lt;/label&gt;

&lt;input type="radio" id="high" name="urgency" value="high"&gt;
&lt;label for="high"&gt;Высокая&lt;/label&gt;
&lt;/div&gt;
&lt;/div&gt;

&lt;div class="form-group"&gt;
&lt;input type="checkbox" id="copy" name="send_copy"&gt;
&lt;label for="copy"&gt;Отправить копию на email&lt;/label&gt;
&lt;/div&gt;</pre>
```

```
<button type="submit">Отправить сообщение</button>
</form>
</div>

<div class="demo-form">
<h3>Демонстрация формы обратной связи:</h3>
<form style="background: white; padding: 20px; border-radius: 5px;">
<div class="form-group">
<label for="demo-subject">Тема сообщения:</label>
<select id="demo-subject" name="subject" required style="width: 100%; padding: 8px;">
<option value="">Выберите тему</option>
<option value="question">Вопрос</option>
<option value="complaint">Жалоба</option>
<option value="suggestion">Предложение</option>
</select>
</div>

<div class="form-group">
<label for="demo-message">Ваше сообщение:</label>
<textarea id="demo-message" name="message" rows="5" placeholder="Опишите Вашу проблему или вопрос..." required style="width: 100%; padding: 8px;"></textarea>
</div>

<div class="form-group">
<label>Срочность:</label>
<div class="radio-group">
<input type="radio" id="demo-low" name="urgency" value="low">
<label for="demo-low" style="display: inline; font-weight: normal;">Низкая</label>

<input type="radio" id="demo-medium" name="urgency" value="medium" checked>
<label for="demo-medium" style="display: inline; font-weight: normal;">Средняя</label>

<input type="radio" id="demo-high" name="urgency" value="high">
<label for="demo-high" style="display: inline; font-weight: normal;">Высокая</label>
</div>
</div>

<div class="form-group">
```

```
<input type="checkbox" id="demo-copy" name="send_copy">
<label for="demo-copy" style="display: inline; font-weight:
normal;">Отправить копию на email</label>
</div>

<button type="submit" style="background: #3498db; color: white; padding:
10px 20px; border: none; border-radius: 3px; cursor: pointer;">Отправить
сообщение</button>
</form>
</div>
</div>

<!-- Шпаргалка по элементам формы --&gt;
&lt;div class="example-section"&gt;
&lt;h2&gt;📋 Шпаргалка по элементам формы&lt;/h2&gt;

&lt;table border="1" style="border-collapse: collapse; width: 100%; margin:
15px 0;"&gt;
&lt;tr style="background-color: #34495e; color: white;"&gt;
&lt;th style="padding: 12px;">Элемент</th>
<th style="padding: 12px;">Код</th>
<th style="padding: 12px;">Назначение</th>
</tr>
<tr>
<td style="padding: 10px;">Текстовое поле</td>
<td style="padding: 10px;"><input type="text"></code>Ввод текста (имя, фамилия)</td>
</tr>
<tr style="background-color: #f9f9f9;">
<td style="padding: 10px;">Поле email</td>
<td style="padding: 10px;"><input type="email"></code>Ввод email с проверкой</td>
</tr>
<tr>
<td style="padding: 10px;">Пароль</td>
<td style="padding: 10px;"><input type="password"></code>Ввод пароля (скрывается)</td>
</tr>
<tr style="background-color: #f9f9f9;">
<td style="padding: 10px;">Числовое поле</td>
<td style="padding: 10px;"><input type="number"></code>Ввод чисел (возраст, количество)</td>
</tr>
<tr>
<td style="padding: 10px;">Выпадающий список</td>
```

```
<td style="padding: 10px;"><code><select><option></select></code></td>
<td style="padding: 10px;">Выбор из нескольких вариантов</td>
</tr>
<tr style="background-color: #f9f9f9;">
<td style="padding: 10px;">Текстовая область</td>
<td style="padding: 10px;"><code><textarea></textarea></code></td>
<td style="padding: 10px;">Многострочный текст (сообщения)</td>
</tr>
<tr>
<td style="padding: 10px;">Радиокнопки</td>
<td style="padding: 10px;"><code><input type="radio"></code></td>
<td style="padding: 10px;">Выбор одного варианта из нескольких</td>
</tr>
<tr style="background-color: #f9f9f9;">
<td style="padding: 10px;">Чекбоксы</td>
<td style="padding: 10px;"><code><input type="checkbox"></code></td>
<td style="padding: 10px;">Выбор нескольких вариантов</td>
</tr>
<tr>
<td style="padding: 10px;">Кнопка отправки</td>
<td style="padding: 10px;"><code><input type="submit"></code></td>
<td style="padding: 10px;">Отправка формы на сервер</td>
</tr>
</table>
</div>
```

```
<!-- Практическое задание -->
<div class="example-section">
<h2>🎯 Практическое задание</h2>

<div class="practice-task">
<h3>Создайте форму "Запись к врачу"</h3>

<div class="code-example">
<!DOCTYPE html>
<html>
<head>
<title>Запись к врачу</title>
</head>
<body>
<h1>Запись на приём к врачу</h1>

<!-- Ваша ЗАДАЧА: Создать форму ниже -->

<form action="/appointment" method="POST">
```

```
<!-- Добавьте поля для: -->
<!-- 1. ФИО пациента -->
<!-- 2. Дата рождения -->
<!-- 3. Номер полиса -->
<!-- 4. Выбор врача (терапевт, хирург, окулист) -->
<!-- 5. Желаемая дата приёма -->
<!-- 6. Время приёма (утро/день) -->
<!-- 7. Контактный телефон -->
<!-- 8. Кнопку "Записаться" -->
</form>
</body>
</html>
</div>
```

```
<div class="note">
<strong> Задание:</strong> Создайте форму записи к врачу, используя изученные элементы:
<ul>
<li>Текстовые поля для ФИО, номера полиса, телефона</li>
<li>Поле даты для даты рождения и даты приёма</li>
<li>Выпадающий список для выбора врача</li>
<li>Радиокнопки для выбора времени приёма</li>
<li>Кнопку отправки формы</li>
<li>Обязательные поля (required)</li>
</ul>
</div>
</div>
</div>
</body>
</html>
```

Часть 5: Что Вы теперь умеете

- Создавать базовую структуру формы
- Использовать различные типы полей ввода
- Добавлять текстовые поля, пароли, email
- Создавать выпадающие списки и радиокнопки
- Работать с чекбоксами и текстовыми областями
- Добавлять кнопки отправки и сброса формы

Часть 6: Итоги и рекомендации

-  **Ключевые моменты:**

Форма начинается с `<form>` — это контейнер для всех полей

Атрибут `action` — указывает, куда отправлять данные

Атрибут `method` — определяет способ отправки (GET/POST)

Каждое поле имеет `name` — это имя для обработки на сервере

`<label>` **связан с** `id` — улучшает доступность формы

Практические советы:

Используйте `required` для обязательных полей

Добавляйте `placeholder` для подсказок в полях

Группируйте поля с помощью `<fieldset>` и `<legend>`

Проверяйте `email` с `type="email"`

Используйте правильные типы полей для разных данных

Следующие шаги:

Изучить валидацию форм (проверка правильности ввода)

Освоить CSS для стилизации форм

Узнать о JavaScript для динамических форм

Понять, как обрабатывать формы на сервере

Помните: Хорошо сделанная форма — это залог удобства пользователей и успешного взаимодействия с Вашим сайтом! 

11.2. Поле для ввода текста <input type="text">

Введение: Основной инструмент для ввода информации

Текстовое поле — это самый распространённый и универсальный элемент формы. Представьте, что это чистая строка в анкете, куда можно вписать любую текстовую информацию: имя, фамилию, адрес, название города или любые другие данные. Сегодня мы подробно изучим все возможности этого элемента!

Часть 1: Базовая структура текстового поля

Самый простой способ создать текстовое поле:

```
html  
<input type="text">
```

Но чтобы поле было полезным, добавим основные атрибуты:

```
html  
<input type="text" name="username" id="username" placeholder="Введите  
ваше имя">
```

Результат:

text
[Введите ваше имя]

Часть 2: Основные атрибуты текстового поля

2.1. Атрибут name — имя поля

```
html  
<input type="text" name="first_name">  
<input type="text" name="last_name">
```

Зачем нужен: Когда форма отправляется на сервер, именно по name сервер понимает, какое поле что означает.

2.2. Атрибут placeholder — подсказка

```
html  
<input type="text" placeholder="Введите Ваш город">
```

Результат:

text

[Введите Ваш город]

Особенности:

Исчезает при начале ввода
Не заменяет label (метку)
Только подсказка, а не значение

2.3. Атрибут value — начальное значение

html
<input type="text" value="Москва" name="city">

Результат:

text
[Москва]

Зачем нужен: Для предзаполненных полей или отображения текущих данных.

2.4. Атрибут required — обязательное поле

html
<input type="text" name="email" required>

Зачем нужен: Браузер не отправит форму, пока обязательное поле не заполнено.

Часть 3: Практические примеры использования

Пример 1: Форма регистрации

html
<form>
<label for="fullname">ФИО:</label>
<input type="text" id="fullname" name="fullname"
placeholder="Иванов Иван Иванович" required>

<label for="city">Город:</label>
<input type="text" id="city" name="city"
placeholder="Москва">

<label for="address">Адрес:</label>
<input type="text" id="address" name="address"
placeholder="ул. Примерная, д. 1, кв. 1">

```
</form>
```

Пример 2: Поиск по сайту

```
html
<form action="/search" method="GET">
<input type="text" name="query"
placeholder="Поиск по сайту..."
style="width: 300px; padding: 10px;">
<button type="submit">Найти</button>
</form>
```

Результат:

```
text
[Поиск по сайту...] [Найти]
```

Часть 4: Дополнительные атрибуты для улучшения функциональности

4.1. Ограничение длины maxlength

```
html
<label>Номер телефона (10 цифр):</label>
<input type="text" name="phone" maxlength="10"
placeholder="9123456789">
```

4.2. Минимальная и максимальная длина minlength

```
html
<label>Пароль (не менее 6 символов):</label>
<input type="text" name="password" minlength="6"
placeholder="Введите пароль">
```

4.3. Размер поля size

```
html
<label>Короткий код:</label>
<input type="text" name="code" size="5" maxlength="5">

<label>Длинный текст:</label>
<input type="text" name="description" size="50">
```

4.4. Только для чтения readonly

```
html
```

```
<label>ID пользователя:</label>
<input type="text" name="user_id" value="12345" readonly>
```

4.5. Отключенное поле disabled

```
html
<label>Премиум функция:</label>
<input type="text" name="premium_feature"
value="Доступно в премиум версии" disabled>
```

Часть 5: Полная страница с примерами и практикой

```
html
<!DOCTYPE html>
<html lang="ru">
<head>
<title>Практика: Текстовые поля в HTML</title>
<meta charset="utf-8">
<style>
body {
font-family: 'Arial', sans-serif;
max-width: 1000px;
margin: 0 auto;
padding: 20px;
line-height: 1.6;
background-color: #f8f9fa;
}
.example-section {
background: white;
padding: 25px;
margin: 25px 0;
border-radius: 10px;
box-shadow: 0 2px 5px rgba(0,0,0,0.1);
}
h1, h2 {
color: #2c3e50;
}
h1 {
border-bottom: 3px solid #3498db;
padding-bottom: 15px;
}
.code-example {
background: #2d2d2d;
color: #f8f8f2;
padding: 15px;
```

```
border-radius: 5px;  
font-family: 'Courier New', monospace;  
margin: 15px 0;  
overflow-x: auto;  
}  
.note {  
background: #e7f3ff;  
padding: 15px;  
border-left: 4px solid #2196F3;  
margin: 15px 0;  
border-radius: 5px;  
}  
.warning {  
background: #fff3e0;  
padding: 15px;  
border-left: 4px solid #ff9800;  
margin: 15px 0;  
border-radius: 5px;  
}
```

/ Стили для демонстрационных форм */*

```
.demo-form {  
background: #f8f9fa;  
padding: 20px;  
border-radius: 8px;  
margin: 20px 0;  
}  
.form-group {  
margin-bottom: 15px;  
}  
.label {  
display: block;  
margin-bottom: 5px;  
font-weight: bold;  
color: #2c3e50;  
}  
input[type="text"] {  
width: 100%;  
padding: 12px;  
border: 2px solid #ddd;  
border-radius: 5px;  
font-size: 16px;  
box-sizing: border-box;  
transition: border-color 0.3s;  
}
```

```
input[type="text"]:focus {  
border-color: #3498db;  
outline: none;  
box-shadow: 0 0 5px rgba(52, 152, 219, 0.3);  
}  
input[type="text"]:invalid {  
border-color: #e74c3c;  
}  
input[type="text"]:valid {  
border-color: #27ae60;  
}  
.required::after {  
content: " *";  
color: #e74c3c;  
}  
.attribute-table {  
width: 100%;  
border-collapse: collapse;  
margin: 20px 0;  
}  
.attribute-table th, .attribute-table td {  
border: 1px solid #ddd;  
padding: 12px;  
text-align: left;  
}  
.attribute-table th {  
background: #34495e;  
color: white;  
}  
.attribute-table tr:nth-child(even) {  
background: #f9f9f9;  
}  
</style>  
</head>  
<body>
```

Текстовые поля в HTML: Универсальный инструмент для ввода данных

```
<div class="note">  
<strong>💡 Основная идея:</strong> Текстовое поле <code><input type="text"></code> — это основной способ получения текстовой информации от пользователя. От простого имени до сложных адресов — этот элемент справляется с любыми текстовыми данными.  
</div>
```

```
<!-- Пример 1: Базовая структура -->
<div class="example-section">
<h2>1. Базовая структура текстового поля</h2>

<div class="code-example">
<!-- Простейшее текстовое поле -->
<input type="text">

<!-- Текстовое поле с основными атрибутами -->
<label for="username">Имя пользователя:</label>
<input type="text"
id="username"
name="username"
placeholder="Введите ваше имя"
value=""
required>
</div>

<div class="demo-form">
<h3>Демонстрация:</h3>
<div style="background: white; padding: 15px; border-radius: 5px;">
<label for="demo-basic">Простое поле:</label>
<input type="text" id="demo-basic" style="width: 100%; padding: 8px; margin-bottom: 15px;">

<label for="demo-advanced" class="required">Поле с атрибутами:</label>
<input type="text" id="demo-advanced" name="username"
placeholder="Введите ваше имя" required style="width: 100%; padding: 8px;">
</div>
</div>
</div>

<!-- Пример 2: Основные атрибуты -->
<div class="example-section">
<h2>2. Основные атрибуты текстового поля</h2>

<table class="attribute-table">
<tr>
<th>Атрибут</th>
<th>Назначение</th>
<th>Пример</th>
<th>Результат</th>
</tr>
<tr>
```

```
<td><code>name</code></td>
<td>Имя поля для обработки на сервере</td>
<td><code>name="first_name"</code></td>
<td>Сервер получит: first_name=Введённое значение</td>
</tr>
<tr>
<td><code>placeholder</code></td>
<td>Подсказка внутри поля</td>
<td><code>placeholder="Ваше имя"</code></td>
<td><input type="text" placeholder="Ваше имя" style="padding: 5px; width: 200px;"></td>
</tr>
<tr>
<td><code>value</code></td>
<td>Начальное значение поля</td>
<td><code>value="Москва"</code></td>
<td><input type="text" value="Москва" style="padding: 5px; width: 200px;"></td>
</tr>
<tr>
<td><code>required</code></td>
<td>Обязательное для заполнения</td>
<td><code>required</code></td>
<td>Браузер требует заполнения</td>
</tr>
<tr>
<td><code>maxlength</code></td>
<td>Максимальное количество символов</td>
<td><code>maxlength="10"</code></td>
<td>Нельзя ввести больше 10 символов</td>
</tr>
<tr>
<td><code>readonly</code></td>
<td>Только для чтения</td>
<td><code>readonly</code></td>
<td><input type="text" value="Нельзя изменить" readonly style="padding: 5px; width: 200px;"></td>
</tr>
<tr>
<td><code>disabled</code></td>
<td>Отключенное поле</td>
<td><code>disabled</code></td>
<td><input type="text" value="Отключено" disabled style="padding: 5px; width: 200px;"></td>
</tr>
```

```
</table>
</div>

<!-- Пример 3: Практические примеры использования --&gt;
&lt;div class="example-section"&gt;
&lt;h2&gt;3. Практические примеры использования&lt;/h2&gt;

&lt;div class="code-example"&gt;
<!-- Пример 1: Форма регистрации --&gt;
&lt;form action="/register" method="POST"&gt;
&lt;div class="form-group"&gt;
&lt;label for="first_name" class="required"&gt;Имя:&lt;/label&gt;
&lt;input type="text" id="first_name" name="first_name"
placeholder="Иван" required&gt;
&lt;/div&gt;

&lt;div class="form-group"&gt;
&lt;label for="last_name" class="required"&gt;Фамилия:&lt;/label&gt;
&lt;input type="text" id="last_name" name="last_name"
placeholder="Иванов" required&gt;
&lt;/div&gt;

&lt;div class="form-group"&gt;
&lt;label for="city"&gt;Город:&lt;/label&gt;
&lt;input type="text" id="city" name="city"
placeholder="Москва"&gt;
&lt;/div&gt;
&lt;/form&gt;

&lt;!-- Пример 2: Поле с ограничением длины --&gt;
&lt;div class="form-group"&gt;
&lt;label for="phone"&gt;Телефон (10 цифр):&lt;/label&gt;
&lt;input type="text" id="phone" name="phone"
placeholder="9123456789"
maxlength="10"
pattern="[0-9]{10}"&gt;
&lt;/div&gt;

&lt;!-- Пример 3: Поле с подсказкой и значением по умолчанию --&gt;
&lt;div class="form-group"&gt;
&lt;label for="country"&gt;Страна:&lt;/label&gt;
&lt;input type="text" id="country" name="country"
value="Россия"
placeholder="Введите страну проживания"&gt;
&lt;/div&gt;</pre>
```

```
</div>

<div class="demo-form">
<h3>Демонстрация практических примеров:</h3>
<form style="background: white; padding: 20px; border-radius: 5px;">
<div class="form-group">
<label for="demo-firstname" class="required">Имя:</label>
<input type="text" id="demo-firstname" name="first_name"
placeholder="Иван" required style="width: 100%; padding: 8px;">
</div>

<div class="form-group">
<label for="demo-lastname" class="required">Фамилия:</label>
<input type="text" id="demo-lastname" name="last_name"
placeholder="Иванов" required style="width: 100%; padding: 8px;">
</div>

<div class="form-group">
<label for="demo-city">Город:</label>
<input type="text" id="demo-city" name="city" placeholder="Москва"
style="width: 100%; padding: 8px;">
</div>

<div class="form-group">
<label for="demo-phone">Телефон (10 цифр):</label>
<input type="text" id="demo-phone" name="phone"
placeholder="9123456789" maxlength="10" style="width: 100%; padding:
8px;">
</div>

<div class="form-group">
<label for="demo-country">Страна:</label>
<input type="text" id="demo-country" name="country" value="Россия"
placeholder="Введите страну проживания" style="width: 100%; padding:
8px;">
</div>
</form>
</div>
</div>

<!-- Пример 4: Специализированные текстовые поля --&gt;
&lt;div class="example-section"&gt;
&lt;h2&gt;4. Специализированные типы текстовых полей&lt;/h2&gt;

&lt;div class="code-example"&gt;</pre>
```

```
<!-- Поле для поиска -->
<input type="search"
name="query"
placeholder="Поиск..."
results="5">

<!-- Поле для телефона -->
<input type="tel"
name="phone"
placeholder="+7 (123) 456-78-90"
pattern="[0-9]{10}">

<!-- Поле для URL -->
<input type="url"
name="website"
placeholder="https://example.com">

<!-- Поле с автозаполнением -->
<input type="text"
name="city"
list="cities"
placeholder="Выберите город">
<datalist id="cities">
<option value="Москва">
<option value="Санкт-Петербург">
<option value="Екатеринбург">
<option value="Новосибирск">
</datalist>
</div>

<div class="demo-form">
<h3>Демонстрация специализированных полей:</h3>
<form style="background: white; padding: 20px; border-radius: 5px;">
<div class="form-group">
<label>Поиск:</label>
<input type="search" placeholder="Поиск..." results="5" style="width: 100%; padding: 8px;">
</div>

<div class="form-group">
<label>Телефон:</label>
<input type="tel" placeholder="+7 (123) 456-78-90" style="width: 100%; padding: 8px;">
</div>
```

```
<div class="form-group">
<label>Веб-сайт:</label>
<input type="url" placeholder="https://example.com" style="width: 100%; padding: 8px;">
</div>

<div class="form-group">
<label>Город (с подсказками):</label>
<input type="text" list="demo-cities" placeholder="Выберите город" style="width: 100%; padding: 8px;">
<datalist id="demo-cities">
<option value="Москва">
<option value="Санкт-Петербург">
<option value="Екатеринбург">
<option value="Новосибирск">
<option value="Казань">
</datalist>
</div>
</form>
</div>
</div>
```

<!-- Пример 5: Валидация и ограничения -->

```
<div class="example-section">
<h2>5. Валидация и ограничения ввода</h2>
```

```
<div class="code-example">
<!-- Обязательное поле -->
<input type="text" name="username" required>

<!-- Ограничение длины -->
<input type="text" name="zipcode" maxlength="6" minlength="6">

<!-- Регулярное выражение для проверки -->
<input type="text" name="phone" pattern="[0-9]{10}">

<!-- Запрет автозаполнения -->
<input type="text" name="secret_code" autocomplete="off">

<!-- Проверка на стороне клиента -->
<input type="text" name="email"
pattern="[a-zA-Z0-9._%+-]+@[a-zA-Z0-9.-]+\.[a-zA-Z]{2,}$"
title="Введите корректный email">
</div>
```

```
<div class="demo-form">
<h3>Демонстрация валидации:</h3>
<form style="background: white; padding: 20px; border-radius: 5px;">
<div class="form-group">
<label class="required">Обязательное поле:</label>
<input type="text" required placeholder="Нельзя оставить пустым"
style="width: 100%; padding: 8px;">
</div>

<div class="form-group">
<label>Почтовый индекс (6 цифр):</label>
<input type="text" maxlength="6" minlength="6" pattern="[0-9]{6}"
placeholder="123456" style="width: 100%; padding: 8px;">
</div>

<div class="form-group">
<label>Без автозаполнения:</label>
<input type="text" autocomplete="off" placeholder="Браузер не запомнит"
style="width: 100%; padding: 8px;">
</div>

<div class="form-group">
<label>Email с проверкой:</label>
<input type="text" pattern="[a-zA-Z._%+-]+@[a-zA-Z.-]+\.[a-zA-Z]{2,}$"
title="Введите корректный email" placeholder="test@example.com"
style="width: 100%; padding: 8px;">
</div>

<button type="submit" style="background: #3498db; color: white; padding:
10px 20px; border: none; border-radius: 3px; cursor: pointer;">Проверить
валидацию</button>
</form>
</div>
</div>

<!-- Шпаргалка по атрибутам --&gt;
&lt;div class="example-section"&gt;
&lt;h2&gt;📋 Шпаргалка по атрибутам текстового поля&lt;/h2&gt;

&lt;div style="display: grid; grid-template-columns: repeat(auto-fit,
minmax(300px, 1fr)); gap: 15px; margin: 20px 0;"&gt;
&lt;div style="border: 2px solid #3498db; padding: 15px; border-radius: 8px;"&gt;
&lt;h3 style="color: #3498db; margin-top: 0;"&gt;🎯 Основные атрибуты&lt;/h3&gt;
&lt;ul&gt;
&lt;li&gt;&lt;code&gt;name&lt;/code&gt; - имя для сервера&lt;/li&gt;</pre>
```

```
<li><code>id</code> - уникальный идентификатор</li>
<li><code>placeholder</code> - подсказка</li>
<li><code>value</code> - значение по умолчанию</li>
<li><code>required</code> - обязательное поле</li>
</ul>
</div>
```

```
<div style="border: 2px solid #27ae60; padding: 15px; border-radius: 8px;">
<h3 style="color: #27ae60; margin-top: 0;">  Атрибуты размера</h3>
<ul>
<li><code>size</code> - видимая ширина</li>
<li><code>maxlength</code> - макс. символов</li>
<li><code>minlength</code> - мин. символов</li>
<li><code>width</code> - ширина через CSS</li>
</ul>
</div>
```

```
<div style="border: 2px solid #e74c3c; padding: 15px; border-radius: 8px;">
<h3 style="color: #e74c3c; margin-top: 0;">  Атрибуты доступа</h3>
<ul>
<li><code>readonly</code> - только чтение</li>
<li><code>disabled</code> - отключено</li>
<li><code>autocomplete</code> - автозаполнение</li>
<li><code>autofocus</code> - автофокус</li>
</ul>
</div>
```

```
<div style="border: 2px solid #f39c12; padding: 15px; border-radius: 8px;">
<h3 style="color: #f39c12; margin-top: 0;">  Атрибуты валидации</h3>
<ul>
<li><code>pattern</code> - регулярное выражение</li>
<li><code>title</code> - подсказка при ошибке</li>
<li><code>novalidate</code> - отключить проверку</li>
</ul>
</div>
</div>
</div>
```

```
<!-- Практическое задание -->
<div class="example-section">
<h2>  Практическое задание</h2>
```

```
<div class="practice-task">
<h3>Создайте форму "Анкета пенсионера"</h3>
```

```
<div class="code-example">
<!DOCTYPE html>
<html>
<head>
<title>Анкета пенсионера</title>
</head>
<body>
<h1>Анкета для получения льгот</h1>

<form action="/submit_pension" method="POST">
<!-- Ваша ЗАДАЧА: Создать поля ниже -->

<!-- 1. ФИО (обязательное) -->
<!-- 2. Дата рождения (только чтение, предзаполненное) -->
<!-- 3. Номер пенсионного удостоверения (макс 20 символов) -->
<!-- 4. Адрес проживания -->
<!-- 5. Телефон (только цифры, макс 11 символов) -->
<!-- 6. Email с проверкой формата -->
<!-- 7. Размер пенсии (только числа) -->

<button type="submit">Отправить анкету</button>
</form>
</body>
</html>
</div>

<div class="note">
<strong>📝 Задание:</strong> Создайте форму анкеты, используя
изученные атрибуты:
<ul>
<li><code>required</code> для обязательных полей</li>
<li><code>readonly</code> для неизменяемых данных</li>
<li><code>maxlength</code> для ограничения длины</li>
<li><code>pattern</code> для проверки формата</li>
<li><code>placeholder</code> для подсказок</li>
<li><code>value</code> для предзаполненных полей</li>
</ul>
</div>
</div>
```

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

Часть 6: Что Вы теперь умеете

- Создавать базовые текстовые поля
- Использовать основные атрибуты (`name`, `placeholder`, `value`)
- Добавлять обязательные поля с `required`
- Ограничивать длину ввода с `maxlength` и `minlength`
- Создавать поля только для чтения и отключенные
- Использовать валидацию с `pattern`
- Работать со специализированными полями (`search`, `tel`, `url`)

Часть 7: Итоги и рекомендации

Ключевые принципы работы с текстовыми полями:

Всегда используйте `name` — без него данные не отправятся на сервер
Добавляйте `placeholder` — помогает пользователю понять, что вводить
Размечайте обязательные поля `required` — предотвращает ошибки
Ограничивайте длину `maxlength` — защищает от переполнения
Используйте правильные типы — улучшает пользовательский опыт

Профессиональные советы:

Сочетайте `<label>` с `id` — улучшает доступность
Используйте `autocomplete` — ускоряет заполнение форм
Добавляйте `title` — поясняет требования к вводу
Тестируйте на мобильных — убедитесь в удобстве ввода
Группируйте связанные поля — используйте `<fieldset>`

Следующие шаги:

Изучить другие типы полей ввода (`email`, `password`, `number`)
Освоить CSS для красивого оформления полей
Узнать о JavaScript для динамической валидации
Понять, как обрабатывать данные формы на сервере

Помните: Хорошо настроенное текстовое поле — это залог корректных данных и довольных пользователей! 

11.3. Кнопка "Отправить" <input type="submit">

Введение: Завершающий шаг в диалоге с сайтом

Представьте, что Вы заполнили бумажную анкету — теперь нужно отнести её в отдел кадров. Кнопка "Отправить" в форме выполняет exactly эту роль! Это важнейший элемент, который завершает процесс заполнения формы и отправляет все данные на сервер для обработки.

Часть 1: Базовая структура кнопки отправки

Самая простая кнопка отправки:

```
html  
<input type="submit">
```

Но чтобы она была понятной пользователю, добавим значение:

```
html  
<input type="submit" value="Отправить">
```

Результат:

```
text  
[ Отправить ]
```

Часть 2: Основные атрибуты кнопки отправки

2.1. Атрибут value — текст на кнопке

```
html  
<input type="submit" value="Отправить данные">  
<input type="submit" value="Зарегистрироваться">  
<input type="submit" value="Войти">
```

Результат:

```
text  
[ Отправить данные ] [ Зарегистрироваться ] [ Войти ]
```

2.2. Атрибут name — имя кнопки

```
html  
<input type="submit" name="submit_btn" value="Отправить">
```

Зачем нужен: Полезно, когда в форме несколько кнопок отправки с разными действиями.

2.3. Атрибут form — связь с формой

```
html
<form id="myForm">
<!-- Поля формы --&gt;
&lt;/form&gt;

&lt;input type="submit" value="Отправить" form="myForm"&gt;</pre>
```

Зачем нужен: Когда кнопка находится вне формы, но должна её отправлять.

Часть 3: Практические примеры использования

Пример 1: Простая контактная форма

```
html
<form action="/contact" method="POST">
<label>Имя:</label>
<input type="text" name="name" required>

<label>Сообщение:</label>
<textarea name="message" required></textarea>

<input type="submit" value="Отправить сообщение">
</form>
```

Пример 2: Форма с несколькими кнопками

```
html
<form action="/order" method="POST">
<!-- Данные заказа --&gt;

&lt;input type="submit" name="action" value="Сохранить черновик"&gt;
&lt;input type="submit" name="action" value="Отправить заказ"&gt;
&lt;/form&gt;</pre>
```

Пример 3: Кнопка с иконкой (используя CSS)

```
html
<input type="submit" value="✉ Отправить" class="icon-button">

css
```

```
.icon-button {  
background-color: #4CAF50;  
color: white;  
padding: 12px 24px;  
border: none;  
border-radius: 4px;  
cursor: pointer;  
font-size: 16px;  
}
```

Часть 4: Альтернативные способы создания кнопки отправки

4.1. Использование тега <button>

```
html  
<button type="submit">  
<strong>Отправить</strong><br>  
<small>Нажмите для подтверждения</small>  
</button>
```

Преимущества: Можно добавлять HTML-разметку внутрь кнопки.

4.2. Кнопка с изображением

```
html  
<button type="submit">  
  
Отправить форму  
</button>
```

Часть 5: Полная страница с примерами и практикой

```
html  
<!DOCTYPE html>  
<html lang="ru">  
<head>  
<title>Практика: Кнопка "Отправить" в HTML</title>  
<meta charset="utf-8">  
<style>  
body {  
font-family: 'Arial', sans-serif;  
max-width: 1000px;  
margin: 0 auto;  
padding: 20px;  
line-height: 1.6;
```

```
background-color: #f8f9fa;
}

.example-section {
background: white;
padding: 25px;
margin: 25px 0;
border-radius: 10px;
box-shadow: 0 2px 5px rgba(0,0,0,0.1);
}

h1, h2 {
color: #2c3e50;
}
h1 {
border-bottom: 3px solid #3498db;
padding-bottom: 15px;
}
.code-example {
background: #2d2d2d;
color: #f8f8f2;
padding: 15px;
border-radius: 5px;
font-family: 'Courier New', monospace;
margin: 15px 0;
overflow-x: auto;
}
.note {
background: #e7f3ff;
padding: 15px;
border-left: 4px solid #2196F3;
margin: 15px 0;
border-radius: 5px;
}
.warning {
background: #fff3e0;
padding: 15px;
border-left: 4px solid #ff9800;
margin: 15px 0;
border-radius: 5px;
}
```

```
/* Стили для демонстрационных форм */
.demo-form {
background: #f8f9fa;
padding: 20px;
border-radius: 8px;
```

```
margin: 20px 0;
}

.form-group {
margin-bottom: 15px;
}
label {
display: block;
margin-bottom: 5px;
font-weight: bold;
color: #2c3e50;
}
input[type="text"],
input[type="email"],
textarea {
width: 100%;
padding: 12px;
border: 2px solid #ddd;
border-radius: 5px;
font-size: 16px;
box-sizing: border-box;
}

/* Базовые стили для кнопок отправки */
input[type="submit"] {
background: #3498db;
color: white;
padding: 12px 25px;
border: none;
border-radius: 5px;
cursor: pointer;
font-size: 16px;
margin-right: 10px;
transition: background-color 0.3s;
}
input[type="submit"]:hover {
background: #2980b9;
}
input[type="submit"]:active {
background: #21618c;
transform: translateY(1px);
}

/* Стили для различных типов кнопок */
.btn-primary {
background: #3498db;
```

```
}

.btn-success {
background: #27ae60;
}
.btn-warning {
background: #f39c12;
}
.btn-danger {
background: #e74c3c;
}
.btn-large {
padding: 15px 30px;
font-size: 18px;
}
.btn-small {
padding: 8px 16px;
font-size: 14px;
}
.btn-block {
display: block;
width: 100%;
}
.btn-with-icon {
padding-left: 45px;
position: relative;
}
.btn-with-icon::before {
content: "✉";
position: absolute;
left: 15px;
top: 50%;
transform: translateY(-50%);
font-size: 18px;
}
```

```
/* Стили для кнопок button */
button[type="submit"] {
background: #9b59b6;
color: white;
padding: 12px 25px;
border: none;
border-radius: 5px;
cursor: pointer;
font-size: 16px;
margin-right: 10px;
```

```
}
```

```
button[type="submit"]:hover {
```

```
background: #8e44ad;
```

```
}
```

```
</style>
```

```
</head>
```

```
<body>
```

```
<h1> Кнопка "Отправить" в HTML: Завершаем процесс заполнения
```

```
формы</h1>
```

```
<div class="note">
```

```
<strong>💡 Основная идея:</strong> Кнопка <code><input
```

```
type="submit"></code> —
```

```
это триггер, который запускает обработку формы. Без неё все
```

```
заполненные данные останутся просто на странице и никогда не
```

```
достигнут сервера.
```

```
</div>
```

```
<!-- Пример 1: Базовая структура -->
```

```
<div class="example-section">
```

```
<h2>1. Базовая структура кнопки отправки</h2>
```

```
<div class="code-example">
```

```
<!-- Простейшая кнопка отправки -->
```

```
<input type="submit">
```

```
<!-- Кнопка с текстом -->
```

```
<input type="submit" value="Отправить">
```

```
<!-- Кнопка с именем -->
```

```
<input type="submit" name="submit_button" value="Отправить данные">
```

```
<!-- Кнопка с формой -->
```

```
<form id="myForm">
```

```
<!-- поля формы -->
```

```
</form>
```

```
<input type="submit" value="Отправить" form="myForm">
```

```
</div>
```

```
<div class="demo-form">
```

```
<h3>Демонстрация:</h3>
```

```
<div style="background: white; padding: 15px; border-radius: 5px;">
```

```
<p>Простая кнопка: <input type="submit" style="padding: 8px 16px;"></p>
```

```
<p>Кнопка с текстом: <input type="submit" value="Отправить" style="padding: 8px 16px;"></p>
<p>Кнопка с именем: <input type="submit" name="submit_button" value="Отправить данные" style="padding: 8px 16px;"></p>
</div>
</div>
</div>
```

<!-- Пример 2: Основные атрибуты -->

```
<div class="example-section">
<h2>2. Основные атрибуты кнопки отправки</h2>
```

Атрибут	Назначение	Пример	Когда использовать
<code>value</code>	Текст на кнопке	<code>value="Отправить"</code>	Всегда! Чтобы пользователь понимал назначение
<code>name</code>	Имя для сервера	<code>name="submit_type"</code>	Когда в форме несколько кнопок с разными действиями
<code>form</code>	Связь с формой	<code>form="form_id"</code>	Когда кнопка находится вне формы

```

<td style="padding: 10px; border: 1px solid #ddd;"><code>disabled</code></td>
<td style="padding: 10px; border: 1px solid #ddd;">Отключение кнопки</td>
<td style="padding: 10px; border: 1px solid #ddd;"><code>disabled</code></td>
<td style="padding: 10px; border: 1px solid #ddd;">Когда форма не
заполнена или обрабатывается</td>
</tr>
</table>
</div>

```

<!-- Пример 3: Практические примеры использования -->

```

<div class="example-section">
<h2>3. Практические примеры использования</h2>

```

```

<div class="code-example">
<!-- Пример 1: Контактная форма -->
<form action="/contact" method="POST">
<div class="form-group">
<label>Ваше имя:</label>
<input type="text" name="name" required>
</div>
<div class="form-group">
<label>Сообщение:</label>
<textarea name="message" required></textarea>
</div>
<input type="submit" value="✉️ Отправить сообщение">
</form>

```

<!-- Пример 2: Форма с выбором действия -->

```

<form action="/document" method="POST">
<!-- поля формы -->
<input type="submit" name="action" value="💾 Сохранить черновик">
<input type="submit" name="action" value="📤 Отправить на проверку">
<input type="submit" name="action" value="📝 Опубликовать">
</form>

```

<!-- Пример 3: Кнопка вне формы -->

```

<form id="searchForm" action="/search" method="GET">
<input type="text" name="query" placeholder="Поиск...">
</form>
<input type="submit" value="🔍 Найти" form="searchForm">
</div>

```

```

<div class="demo-form">

```

```
<h3>Демонстрация практических примеров:</h3>
```

```
<div style="background: white; padding: 20px; border-radius: 5px; margin-bottom: 20px;">
<h4>Контактная форма:</h4>
<form style="margin-bottom: 0;">
<div class="form-group">
<label>Ваше имя:</label>
<input type="text" name="name" required style="width: 100%; padding: 8px;">
</div>
<div class="form-group">
<label>Сообщение:</label>
<textarea name="message" required style="width: 100%; padding: 8px; height: 80px;"></textarea>
</div>
<input type="submit" value="✉️ Отправить сообщение" style="padding: 10px 20px;">
</form>
</div>
```

```
<div style="background: white; padding: 20px; border-radius: 5px; margin-bottom: 20px;">
<h4>Форма с выбором действия:</h4>
<form style="margin-bottom: 0;">
<input type="submit" name="action" value="📝 Сохранить черновик" style="padding: 10px 15px; margin-right: 10px;">
<input type="submit" name="action" value="📤 Отправить на проверку" style="padding: 10px 15px; margin-right: 10px;">
<input type="submit" name="action" value="✅ Опубликовать" style="padding: 10px 15px;">
</form>
</div>
```

```
<div style="background: white; padding: 20px; border-radius: 5px;">
<h4>Кнопка вне формы:</h4>
<form id="demoSearchForm" style="margin-bottom: 10px;">
<input type="text" name="query" placeholder="Поиск..." style="width: 200px; padding: 8px;">
</form>
<input type="submit" value="🔍 Найти" form="demoSearchForm" style="padding: 10px 20px;">
</div>
</div>
</div>
```

```
<!-- Пример 4: Стилизация кнопок отправки -->
<div class="example-section">
<h2>4. Стилизация кнопок отправки с помощью CSS</h2>

<div class="code-example">
<!-- Различные стили кнопок -->
<input type="submit" value="Обычная" class="btn-primary">
<input type="submit" value="Успех" class="btn-success">
<input type="submit" value="Предупреждение" class="btn-warning">
<input type="submit" value="Опасность" class="btn-danger">

<input type="submit" value="Большая кнопка" class="btn-large">
<input type="submit" value="Маленькая" class="btn-small">
<input type="submit" value="На всю ширину" class="btn-block">

<input type="submit" value="Отправить email" class="btn-with-icon">

<!-- Кнопка с тегом button -->
<button type="submit">
<strong>Отправить заявку</strong><br>
<small>На обработку потребуется до 24 часов</small>
</button>
</div>

<div class="demo-form">
<h3>Демонстрация стилизованных кнопок:</h3>
<div style="background: white; padding: 20px; border-radius: 5px;">
<h4>Цветовые варианты:</h4>
<input type="submit" value="Обычная" class="btn-primary" style="margin-bottom: 10px;"><br>
<input type="submit" value="Успех" class="btn-success" style="margin-bottom: 10px;"><br>
<input type="submit" value="Предупреждение" class="btn-warning" style="margin-bottom: 10px;"><br>
<input type="submit" value="Опасность" class="btn-danger" style="margin-bottom: 20px;">

<h4>Размеры:</h4>
<input type="submit" value="Большая кнопка" class="btn-large" style="margin-bottom: 10px;"><br>
<input type="submit" value="Маленькая" class="btn-small" style="margin-bottom: 10px;"><br>
<input type="submit" value="На всю ширину" class="btn-block" style="margin-bottom: 20px;">
```

```
<h4>С иконкой:</h4>
<input type="submit" value="Отправить email" class="btn-with-icon"
style="margin-bottom: 20px;">

<h4>Кнопка с HTML-разметкой:</h4>
<button type="submit" style="background: #9b59b6; color: white; padding:
12px; border: none; border-radius: 5px; cursor: pointer; text-align: left;">
<strong>Отправить заявку</strong><br>
<small>На обработку потребуется до 24 часов</small>
</button>
</div>
</div>
</div>

<!-- Пример 5: Отключенные кнопки и валидация --&gt;
&lt;div class="example-section"&gt;
&lt;h2&gt;5. Отключенные кнопки и работа с валидацией&lt;/h2&gt;

&lt;div class="code-example"&gt;
<!-- Отключенная кнопка --&gt;
&lt;input type="submit" value="Отправить" disabled&gt;

<!-- Кнопка, которая становится активной после валидации --&gt;
&lt;form id="validatedForm" oninput="checkForm()"&gt;
&lt;input type="email" name="email" placeholder="Email" required&gt;
&lt;input type="password" name="password" placeholder="Пароль" required&gt;
&lt;input type="submit" value="Войти" id="submitBtn" disabled&gt;
&lt;/form&gt;

&lt;script&gt;
function checkForm() {
const form = document.getElementById('validatedForm');
const submitBtn = document.getElementById('submitBtn');
// Простая проверка - если форма валидна, включаем кнопку
submitBtn.disabled = !form.checkValidity();
}
&lt;/script&gt;

<!-- Кнопка с подтверждением --&gt;
&lt;input type="submit" value="Удалить аккаунт"
onclick="return confirm('Вы уверены, что хотите удалить аккаунт?')"&gt;
&lt;/div&gt;

&lt;div class="demo-form"&gt;</pre>
```

```
<h3>Демонстрация отключенных кнопок:</h3>
<div style="background: white; padding: 20px; border-radius: 5px;">
<p>Отключенная кнопка: <input type="submit" value="Отправить" disabled
style="padding: 10px 20px;"></p>

<h4>Форма с валидацией:</h4>
<form oninput="demoCheckForm()" style="margin-bottom: 15px;">
<div class="form-group">
<label>Email:</label>
<input type="email" name="email" placeholder="test@example.com" required
style="width: 100%; padding: 8px;">
</div>
<div class="form-group">
<label>Пароль:</label>
<input type="password" name="password" placeholder="Не менее 6
символов" required minlength="6" style="width: 100%; padding: 8px;">
</div>
<input type="submit" value="Войти" id="demoSubmitBtn" disabled
style="padding: 10px 20px;">
</form>

<h4>Кнопка с подтверждением:</h4>
<input type="submit" value="Удалить аккаунт"
onclick="return confirm('Вы уверены, что хотите удалить аккаунт? Это
действие нельзя отменить.')"
style="background: #e74c3c; padding: 10px 20px;">
</div>
</div>

<script>
function demoCheckForm() {
const form = document.querySelector('form[oninput="demoCheckForm()"]');
const submitBtn = document.getElementById('demoSubmitBtn');
submitBtn.disabled = !form.checkValidity();
}
</script>
</div>

<!-- Шпаргалка по кнопкам отправки -->
<div class="example-section">
<h2>📋 Шпаргалка по кнопкам отправки</h2>

<div style="display: grid; grid-template-columns: repeat(auto-fit,
minmax(300px, 1fr)); gap: 15px; margin: 20px 0;">
<div style="border: 2px solid #3498db; padding: 15px; border-radius: 8px;">
```

```
<h3 style="color: #3498db; margin-top: 0;">🎯 Основные способы</h3>
<ul>
<li><code><input type="submit"></code> - стандартная кнопка</li>
<li><code><button type="submit"></code> - кнопка с HTML внутри</li>
<li><code><input type="image"></code> - кнопка-изображение</li>
</ul>
</div>
```

```
<div style="border: 2px solid #27ae60; padding: 15px; border-radius: 8px;">
<h3 style="color: #27ae60; margin-top: 0;">💡 Лучшие практики</h3>
<ul>
<li>Всегда указывайте <code>value</code></li>
<li>Используйте понятный текст</li>
<li>Отключайте при обработке</li>
<li>Добавляйте подтверждение для важных действий</li>
</ul>
</div>
```

```
<div style="border: 2px solid #e74c3c; padding: 15px; border-radius: 8px;">
<h3 style="color: #e74c3c; margin-top: 0;">🚫 Чего избегать</h3>
<ul>
<li>Кнопок без текста</li>
<li>Неочевидных названий</li>
<li>Слишком много кнопок в одной форме</li>
<li>Отправки без подтверждения данных</li>
</ul>
</div>
</div>
</div>
```

```
<!-- Практическое задание -->
<div class="example-section">
<h2>🎯 Практическое задание</h2>
```

```
<div class="practice-task">
<h3>Создайте форму "Онлайн-заявка" с различными кнопками отправки</h3>
```

```
<div class="code-example">
<!DOCTYPE html>
<html>
<head>
<title>Онлайн-заявка</title>
<style>
/* Ваши стили здесь */
```

```
</style>
</head>
<body>
<h1>Онлайн-заявка на консультацию</h1>

<form action="/consultation" method="POST">
<!-- Поля формы --&gt;
&lt;div&gt;
&lt;label&gt;ФИО:&lt;/label&gt;
&lt;input type="text" name="fullname" required&gt;
&lt;/div&gt;

&lt;div&gt;
&lt;label&gt;Телефон:&lt;/label&gt;
&lt;input type="tel" name="phone" required&gt;
&lt;/div&gt;

&lt;div&gt;
&lt;label&gt;Вопрос:&lt;/label&gt;
&lt;textarea name="question" required&gt;&lt;/textarea&gt;
&lt;/div&gt;

<!-- Ваша ЗАДАЧА: Создать кнопки ниже --&gt;

<!-- 1. Основная кнопка отправки --&gt;

<!-- 2. Кнопка "Сохранить черновик" (другое имя) --&gt;

<!-- 3. Отключенная кнопка "Требуется заполнить все поля" --&gt;

<!-- 4. Кнопка с подтверждением для важного действия --&gt;

<!-- 5. Кнопка вне формы, но связанная с ней --&gt;

&lt;/form&gt;

<!-- 6. Кнопка button с разметкой --&gt;

&lt;/body&gt;
&lt;/html&gt;
&lt;/div&gt;

&lt;div class="note"&gt;
&lt;strong&gt;Задание:&lt;/strong&gt; Создайте различные типы кнопок отправки:
</pre>
```

```
<ul>
<li>Основную кнопку с понятным текстом</li>
<li>Альтернативную кнопку с другим действием</li>
<li>Отключенную кнопку с пояснением</li>
<li>Кнопку с подтверждением действия</li>
<li>Кнопку вне формы, но связанную с ней</li>
<li>Кнопку <code><button></code> с HTML-разметкой</li>
</ul>
</div>
</div>
</div>
</body>
</html>
```

Часть 6: Что Вы теперь умеете

- Создавать базовые кнопки отправки
- Использовать атрибуты `value`, `name`, `form`
- Создавать отключенные кнопки
- Работать с несколькими кнопками в одной форме
- Использовать тег `<button>` для сложных кнопок
- Добавлять подтверждение действий
- Стилизовать кнопки с помощью CSS

Часть 7: Итоги и рекомендации

Ключевые принципы работы с кнопками отправки:

Всегда указывайте `value` — пользователь должен понимать, что делает кнопка

Используйте понятные названия — "Отправить", "Сохранить", "Зарегистрироваться"

Отключайте кнопку при обработке — предотвращает повторные отправки

Добавляйте подтверждение для необратимых действий

Тестируйте на мобильных — убедитесь, что кнопка удобна для нажатия

Профессиональные советы:

Используйте `<button>` когда нужна сложная разметка

Добавляйте иконки для лучшей визуализации

Группируйте связанные кнопки — используйте общий контейнер

Проверяйте доступность — убедитесь, что кнопки работают с клавиатурой

Тестируйте разные сценарии — что происходит при двойном нажатии?



Следующие шаги:

Изучить обработку форм на сервере

Освоить JavaScript для динамического управления кнопками

Узнать о AJAX для отправки форм без перезагрузки страницы

Изучить доступность (accessibility) для кнопок

Помните: Хорошо сделанная кнопка отправки — это последнее, что видит пользователь перед успешным завершением действия.
Сделайте этот момент приятным!

11.4. Большое поле для текста <textarea> (для отзывов или писем)

Введение: Пространство для развёрнутых мыслей

Представьте, что Вам нужно написать письмо другу или оставить подробный отзыв о товаре. Обычное текстовое поле для этого не подойдёт — оно слишком короткое! Текстовая область <textarea> создана именно для таких случаев: она позволяет вводить многострочный текст любой длины, идеально подходит для сообщений, комментариев, описаний и писем.

Часть 1: Базовая структура текстовой области

Самая простая текстовая область:

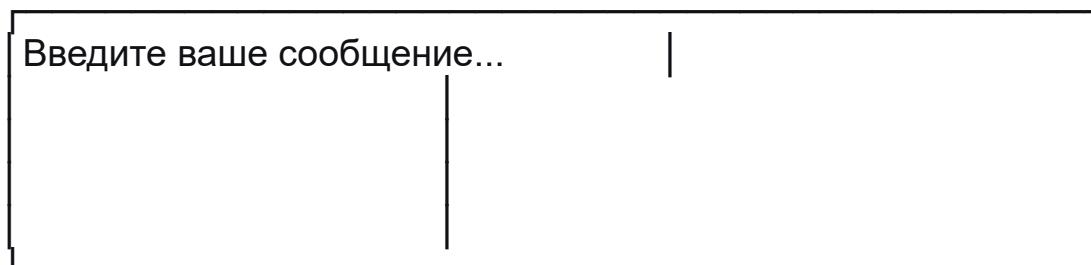
```
html  
<textarea></textarea>
```

Но чтобы она была полезной, добавим основные атрибуты:

```
html  
<textarea name="message" rows="5" cols="40" placeholder="Введите ваше сообщение..."></textarea>
```

Результат:

text



A screenshot of a web browser window showing a single-line text input field. The field has a black border and contains the placeholder text "Введите ваше сообщение..." in a light gray font. A vertical cursor bar is positioned to the right of the placeholder text, indicating where text would be entered.

Часть 2: Основные атрибуты текстовой области

2.1. Атрибуты rows и cols — размеры области

```
html  
<textarea rows="4" cols="50"></textarea>  
<textarea rows="10" cols="30"></textarea>
```

Что означают:

rows — количество видимых строк (высота)

`cols` — количество видимых символов в строке (ширина)

2.2. Атрибут `placeholder` — подсказка

html

```
<textarea placeholder="Опишите Вашу проблему подробно..."></textarea>
```

2.3. Атрибут `name` — имя для сервера

html

```
<textarea name="user_message"></textarea>
```

2.4. Атрибут `maxlength` — ограничение длины

html

```
<textarea maxlength="500" placeholder="Не более 500  
символов"></textarea>
```

Часть 3: Практические примеры использования

Пример 1: Форма обратной связи

html

```
<form action="/feedback" method="POST">  
<label>Ваше имя:</label>  
<input type="text" name="name" required>  
  
<label>Ваш отзыв:</label>  
<textarea name="feedback" rows="6" cols="50"  
placeholder="Поделитесь Вашиими впечатлениями..."  
required></textarea>  
  
<input type="submit" value="Отправить отзыв">  
</form>
```

Пример 2: Поле для длинного описания

html

```
<label>Описание товара:</label>  
<textarea name="description" rows="8" cols="60"  
placeholder="Подробно опишите товар, его особенности и состояние">  
</textarea>
```

Пример 3: Текстовая область с начальным значением

html

```
<textarea name="template" rows="10" cols="40">
```

Уважаемый клиент,

Благодарим Вас за обращение.

Наша служба поддержки свяжется с Вами в ближайшее время.

С уважением,
Команда компании
</textarea>

Часть 4: Полная страница с примерами и практикой

html

```
<!DOCTYPE html>
<html lang="ru">
<head>
<title>Практика: Текстовая область textarea в HTML</title>
<meta charset="utf-8">
<style>
body {
font-family: 'Arial', sans-serif;
max-width: 1000px;
margin: 0 auto;
padding: 20px;
line-height: 1.6;
background-color: #f8f9fa;
}
.example-section {
background: white;
padding: 25px;
margin: 25px 0;
border-radius: 10px;
box-shadow: 0 2px 5px rgba(0,0,0,0.1);
}
h1, h2 {
color: #2c3e50;
}
h1 {
border-bottom: 3px solid #3498db;
padding-bottom: 15px;
}
.code-example {
background: #2d2d2d;
color: #f8f8f2;
padding: 15px;
```

```
border-radius: 5px;  
font-family: 'Courier New', monospace;  
margin: 15px 0;  
overflow-x: auto;  
}  
.note {  
background: #e7f3ff;  
padding: 15px;  
border-left: 4px solid #2196F3;  
margin: 15px 0;  
border-radius: 5px;  
}  
.warning {  
background: #fff3e0;  
padding: 15px;  
border-left: 4px solid #ff9800;  
margin: 15px 0;  
border-radius: 5px;  
}
```

/ Стили для демонстрационных форм */*

```
.demo-form {  
background: #f8f9fa;  
padding: 20px;  
border-radius: 8px;  
margin: 20px 0;  
}  
.form-group {  
margin-bottom: 20px;  
}  
.label {  
display: block;  
margin-bottom: 8px;  
font-weight: bold;  
color: #2c3e50;  
}
```

/ Базовые стили для textarea */*

```
textarea {  
width: 100%;  
padding: 12px;  
border: 2px solid #ddd;  
border-radius: 5px;  
font-size: 16px;  
font-family: inherit;
```

```
box-sizing: border-box;
resize: vertical;
transition: border-color 0.3s, box-shadow 0.3s;
}
textarea:focus {
border-color: #3498db;
outline: none;
box-shadow: 0 0 5px rgba(52, 152, 219, 0.3);
}
textarea:invalid {
border-color: #e74c3c;
}
textarea:valid {
border-color: #27ae60;
}
```

/ Специальные стили для textarea */*

```
.textarea-small {
min-height: 80px;
max-height: 120px;
}
.textarea-medium {
min-height: 120px;
max-height: 200px;
}
.textarea-large {
min-height: 200px;
max-height: 400px;
}
.textarea-no-resize {
resize: none;
}
.textarea-auto-resize {
resize: none;
overflow: hidden;
}
.char-counter {
font-size: 12px;
color: #666;
text-align: right;
margin-top: 5px;
}
.char-counter.warning {
color: #e74c3c;
}
```

```
/* Стили для кнопок */
input[type="submit"] {
background: #3498db;
color: white;
padding: 12px 25px;
border: none;
border-radius: 5px;
cursor: pointer;
font-size: 16px;
}
```

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

```
<body>
```

```
<h1>📝 Текстовая область <textarea>: Пространство для развёрнутых мыслей</h1>
```

```
<div class="note">
<strong>💡 Основная идея:</strong> Текстовая область
<code><textarea></code> — это специальное поле для ввода
многострочного текста. В отличие от обычного текстового поля,
она позволяет писать длинные сообщения, отзывы, комментарии и
письма с переносами строк.
</div>
```

```
<!-- Пример 1: Базовая структура -->
```

```
<div class="example-section">
<h2>1. Базовая структура текстовой области</h2>
```

```
<div class="code-example">
<!-- Простейшая текстовая область -->
<textarea></textarea>
```

```
<!-- Текстовая область с атрибутами размера -->
<textarea rows="5" cols="40"></textarea>
```

```
<!-- Текстовая область с подсказкой и именем -->
<textarea name="message"
rows="6"
cols="50"
placeholder="Введите ваше сообщение..."></textarea>
```

```
<!-- Текстовая область с начальным текстом -->
<textarea name="template" rows="4" cols="40">
Это начальный текст
```

который уже находится
в текстовой области

```
</textarea>  
</div>
```

```
<div class="demo-form">  
<h3>Демонстрация:</h3>  
<div style="background: white; padding: 15px; border-radius: 5px;">  
<div class="form-group">  
<label>Простая текстовая область:</label>  
<textarea></textarea>  
</div>
```

```
<div class="form-group">  
<label>С размерами (5 строк, 40 символов):</label>  
<textarea rows="5" cols="40"></textarea>  
</div>
```

```
<div class="form-group">  
<label>С подсказкой:</label>  
<textarea name="message" rows="4" placeholder="Введите ваше  
сообщение..."></textarea>  
</div>
```

```
<div class="form-group">  
<label>С начальным текстом:</label>  
<textarea name="template" rows="4">  
Это начальный текст  
который уже находится  
в текстовой области  
Можно его отредактировать  
</textarea>  
</div>  
</div>  
</div>  
</div>
```

```
<!-- Пример 2: Основные атрибуты -->  
<div class="example-section">  
<h2>2. Основные атрибуты текстовой области</h2>
```

```
<table style="width: 100%; border-collapse: collapse; margin: 20px 0;">  
<tr style="background-color: #34495e; color: white;">  
<th style="padding: 12px;">Атрибут</th>  
<th style="padding: 12px;">Назначение</th>
```

```
<th style="padding: 12px;">Пример</th>
<th style="padding: 12px;">Результат</th>
</tr>
<tr>
<td style="padding: 10px; border: 1px solid #ddd;"><code>rows</code></td>
<td style="padding: 10px; border: 1px solid #ddd;">Количество видимых
строк</td>
<td style="padding: 10px; border: 1px solid
#ddd;"><code>rows="5"</code></td>
<td style="padding: 10px; border: 1px solid #ddd;">Высота ~5 строк
текста</td>
</tr>
<tr style="background-color: #f9f9f9;">
<td style="padding: 10px; border: 1px solid #ddd;"><code>cols</code></td>
<td style="padding: 10px; border: 1px solid #ddd;">Ширина в символах</td>
<td style="padding: 10px; border: 1px solid
#ddd;"><code>cols="40"</code></td>
<td style="padding: 10px; border: 1px solid #ddd;">Ширина ~40
символов</td>
</tr>
<tr>
<td style="padding: 10px; border: 1px solid #ddd;"><code>name</code></td>
<td style="padding: 10px; border: 1px solid #ddd;">Имя для сервера</td>
<td style="padding: 10px; border: 1px solid
#ddd;"><code>name="message"</code></td>
<td style="padding: 10px; border: 1px solid #ddd;">На сервер отправится
message=текст</td>
</tr>
<tr style="background-color: #f9f9f9;">
<td style="padding: 10px; border: 1px solid
#ddd;"><code>placeholder</code></td>
<td style="padding: 10px; border: 1px solid #ddd;">Подсказка</td>
<td style="padding: 10px; border: 1px solid
#ddd;"><code>placeholder="Опишите..."</code></td>
<td style="padding: 10px; border: 1px solid #ddd;">Исчезает при вводе</td>
</tr>
<tr>
<td style="padding: 10px; border: 1px solid
#ddd;"><code>maxlength</code></td>
<td style="padding: 10px; border: 1px solid #ddd;">Макс. символов</td>
<td style="padding: 10px; border: 1px solid
#ddd;"><code>maxlength="500"</code></td>
<td style="padding: 10px; border: 1px solid #ddd;">Не более 500
символов</td>
</tr>
```

```

<tr style="background-color: #f9f9f9;">
<td style="padding: 10px; border: 1px solid #ddd;"><code>readonly</code></td>
<td style="padding: 10px; border: 1px solid #ddd;">Только чтение</td>
<td style="padding: 10px; border: 1px solid #ddd;"><code>readonly</code></td>
<td style="padding: 10px; border: 1px solid #ddd;">Текст нельзя
изменить</td>
</tr>
<tr>
<td style="padding: 10px; border: 1px solid #ddd;"><code>disabled</code></td>
<td style="padding: 10px; border: 1px solid #ddd;">Отключено</td>
<td style="padding: 10px; border: 1px solid #ddd;"><code>disabled</code></td>
<td style="padding: 10px; border: 1px solid #ddd;">Неактивное поле</td>
</tr>
</table>
</div>

```

<!-- Пример 3: Практические примеры использования -->

```

<div class="example-section">
<h2>3. Практические примеры использования</h2>

```

```

<div class="code-example">
<!-- Пример 1: Форма обратной связи -->
<form action="/feedback" method="POST">
<div class="form-group">
<label for="name">Ваше имя:</label>
<input type="text" id="name" name="name" required>
</div>

```

```

<div class="form-group">
<label for="email">Email:</label>
<input type="email" id="email" name="email" required>
</div>

```

```

<div class="form-group">
<label for="message">Ваше сообщение:</label>
<textarea id="message" name="message"
rows="6"
placeholder="Опишите Вашу проблему или вопрос подробно...">
</div>

```

```
<input type="submit" value="✉️ Отправить сообщение">
</form>
```

```
<!-- Пример 2: Книга отзывов -->
<form action="/guestbook" method="POST">
<div class="form-group">
<label>Ваш отзыв о нашем сервисе:</label>
<textarea name="review"
rows="8"
maxlength="1000"
placeholder="Поделитесь Вашими впечатлениями. Что Вам понравилось?
Что можно улучшить?"></textarea>
<div class="char-counter">Осталось символов: <span
id="counter">1000</span></div>
</div>
<input type="submit" value="⭐️ Оставить отзыв">
</form>
```

```
<!-- Пример 3: Шаблон письма -->
<div class="form-group">
<label>Шаблон ответа клиенту:</label>
<textarea name="email_template" rows="10" readonly>
Уважаемый клиент!
```

Благодарим Вас за обращение в нашу компанию.
Мы получили ваше сообщение и рассмотрим его в ближайшее время.

Если у Вас есть срочный вопрос, Вы можете позвонить нам по телефону:
+7 (123) 456-78-90

С уважением,
Служба поддержки компании "Пример"
</textarea>
</div>
</div>

```
<div class="demo-form">
<h3>Демонстрация практических примеров:</h3>
```

```
<div style="background: white; padding: 20px; border-radius: 5px; margin-
bottom: 20px;">
<h4>Форма обратной связи:</h4>
<form style="margin-bottom: 0;">
<div class="form-group">
<label>Ваше имя:</label>
```

```
<input type="text" name="name" required style="width: 100%; padding: 8px;">
</div>

<div class="form-group">
<label>Email:</label>
<input type="email" name="email" required style="width: 100%; padding: 8px;">
</div>

<div class="form-group">
<label>Ваше сообщение:</label>
<textarea name="message" rows="6" placeholder="Опишите Вашу проблему или вопрос подробно..." required style="width: 100%; padding: 8px;"></textarea>
</div>

<input type="submit" value="✉️ Отправить сообщение" style="padding: 10px 20px;">
</form>
</div>

<div style="background: white; padding: 20px; border-radius: 5px; margin-bottom: 20px;">
<h4>Книга отзывов:</h4>
<form style="margin-bottom: 0;">
<div class="form-group">
<label>Ваш отзыв о нашем сервисе:</label>
<textarea name="review" rows="6" maxlength="200" placeholder="Поделитесь Вашиими впечатлениями. Что Вам понравилось? Что можно улучшить?" style="width: 100%; padding: 8px;"></textarea>
<div class="char-counter">Осталось символов: <span id="demoCounter">200</span></div>
</div>
<input type="submit" value="🌟 Оставить отзыв" style="padding: 10px 20px;">
</form>
</div>

<div style="background: white; padding: 20px; border-radius: 5px;">
<h4>Шаблон письма (только чтение):</h4>
<div class="form-group">
<label>Шаблон ответа клиенту:</label>
<textarea name="email_template" rows="6" readonly style="width: 100%; padding: 8px; background: #f8f9fa;">
```

Уважаемый клиент!

Благодарим Вас за обращение в нашу компанию.

Мы получили ваше сообщение и рассмотрим его в ближайшее время.

Если у Вас есть срочный вопрос, Вы можете позвонить нам по телефону:

+7 (123) 456-78-90

С уважением,

Служба поддержки компании "Пример"

</textarea>

</div>

</div>

</div>

<script>

// Счетчик символов для демо

```
const demoTextarea = document.querySelector("textarea[name='review']");
const demoCounter = document.getElementById('demoCounter');
```

```
demoTextarea.addEventListener('input', function() {
```

```
const remaining = 200 - this.value.length;
```

```
demoCounter.textContent = remaining;
```

```
if (remaining < 50) {
```

```
demoCounter.className = 'char-counter warning';
```

```
} else {
```

```
demoCounter.className = 'char-counter';
```

```
}
```

```
});
```

```
</script>
```

</div>

<!-- Пример 4: Стилизация и дополнительные возможности -->

<div class="example-section">

<h2>4. Стилизация и дополнительные возможности</h2>

<div class="code-example">

<!-- Различные стили текстовых областей -->

<!-- Маленькая текстовая область -->

<textarea class="textarea-small"

placeholder="Короткий комментарий (80-120 символов)"></textarea>

<!-- Средняя текстовая область -->

```
<textarea class="textarea-medium"
placeholder="Обычное сообщение (120-200 символов)"></textarea>

<!-- Большая текстовая область -->
<textarea class="textarea-large"
placeholder="Подробное описание (200+ символов)"></textarea>

<!-- Текстовая область без изменения размера -->
<textarea class="textarea-no-resize"
placeholder="Фиксированный размер"></textarea>

<!-- Автоматическое изменение размера -->
<textarea class="textarea-auto-resize"
placeholder="Размер подстраивается под текст"></textarea>

<style>
.textarea-small {
min-height: 80px;
max-height: 120px;
resize: vertical;
}

.textarea-medium {
min-height: 120px;
max-height: 200px;
resize: vertical;
}

.textarea-large {
min-height: 200px;
max-height: 400px;
resize: vertical;
}

.textarea-no-resize {
resize: none;
}

.textarea-auto-resize {
resize: none;
overflow: hidden;
min-height: 40px;
max-height: 400px;
}
</style>
```

```
</div>

<div class="demo-form">
<h3>Демонстрация стилизованных текстовых областей:</h3>
<div style="background: white; padding: 20px; border-radius: 5px;">
<div class="form-group">
<label>Маленькая область:</label>
<textarea class="textarea-small" placeholder="Короткий комментарий (80-120 символов)" style="min-height: 80px; max-height: 120px; resize: vertical;"></textarea>
</div>

<div class="form-group">
<label>Средняя область:</label>
<textarea class="textarea-medium" placeholder="Обычное сообщение (120-200 символов)" style="min-height: 120px; max-height: 200px; resize: vertical;"></textarea>
</div>

<div class="form-group">
<label>Большая область:</label>
<textarea class="textarea-large" placeholder="Подробное описание (200+ символов)" style="min-height: 200px; max-height: 400px; resize: vertical;"></textarea>
</div>

<div class="form-group">
<label>Без изменения размера:</label>
<textarea class="textarea-no-resize" placeholder="Фиксированный размер (нельзя растянуть)" style="resize: none;"></textarea>
</div>

<div class="form-group">
<label>С автоматическим счетчиком символов:</label>
<textarea id="autoResize" placeholder="Пишите здесь..." style="resize: none; overflow: hidden; min-height: 40px; max-height: 200px;"></textarea>
<div class="char-counter">Символов: <span id="autoCounter">0</span></div>
</div>
</div>
</div>

<script>
// Автоматический счетчик символов
const autoTextarea = document.getElementById('autoResize');
```

```
const autoCounter = document.getElementById('autoCounter');

autoTextarea.addEventListener('input', function() {
  autoCounter.textContent = this.value.length;

  // Автоматическое изменение высоты
  this.style.height = 'auto';
  this.style.height = (this.scrollHeight) + 'px';
});

</script>
</div>
```

<!-- Пример 5: Специальные варианты использования -->

```
<div class="example-section">
<h2>5. Специальные варианты использования</h2>
```

```
<div class="code-example">
<!-- Текстовая область для кода -->
<div class="form-group">
<label>HTML код:</label>
<textarea name="html_code"
rows="10"
placeholder="Введите Ваш HTML код здесь"
style="font-family: 'Courier New', monospace; font-size: 14px;">
<!DOCTYPE html>
<html>
<head>
<title>Пример</title>
</head>
<body>
<h1>Привет, мир!</h1>
</body>
</html>
</textarea>
</div>
```

<!-- Текстовая область с подсказками -->

```
<div class="form-group">
<label>Описание товара:</label>
<textarea name="product_description"
rows="8"
placeholder="Опишите товар по пунктам:
• Состояние товара
• Комплектация
• Особенности">
```

```
• Недостатки (если есть)"></textarea>
</div>

<!-- Текстовая область для заметок --&gt;
&lt;div class="form-group"&gt;
&lt;label&gt;Личные заметки:&lt;/label&gt;
&lt;textarea name="notes"
rows="12"
style="background: #ffffacd; border: 1px dashed #ccc;"
placeholder="Записывайте здесь Ваши мысли и идеи..."&gt;&lt;/textarea&gt;
&lt;/div&gt;
&lt;/div&gt;

&lt;div class="demo-form"&gt;
&lt;h3&gt;Демонстрация специальных вариантов:&lt;/h3&gt;
&lt;div style="background: white; padding: 20px; border-radius: 5px;"&gt;
&lt;div class="form-group"&gt;
&lt;label&gt;Для программирования (моноширинный шрифт):&lt;/label&gt;
&lt;textarea name="html_code" rows="6" placeholder="Введите Ваш код
здесь" style="font-family: 'Courier New', monospace; font-size: 14px; width:
100%; padding: 8px;"&gt;
&lt;!DOCTYPE html&gt;
&lt;html&gt;
&lt;head&gt;
&lt;title&gt;Пример страницы&lt;/title&gt;
&lt;/head&gt;
&lt;body&gt;
&lt;h1&gt;Привет, мир!&lt;/h1&gt;
&lt;p&gt;Это пример HTML кода.&lt;/p&gt;
&lt;/body&gt;
&lt;/html&gt;
&lt;/textarea&gt;
&lt;/div&gt;

&lt;div class="form-group"&gt;
&lt;label&gt;С многострочной подсказкой:&lt;/label&gt;
&lt;textarea name="product_description" rows="6" placeholder="Опишите
товар по пунктам:
• Состояние товара
• Комплектация
• Особенности
• Недостатки (если есть)" style="width: 100%; padding: 8px;"&gt;&lt;/textarea&gt;
&lt;/div&gt;

&lt;div class="form-group"&gt;</pre>
```

```
<label>Поле для заметок (специальный стиль):</label>
<textarea name="notes" rows="5" style="background: #ffffacd; border: 1px
dashed #ccc; width: 100%; padding: 8px;" placeholder="Записывайте здесь
Ваши мысли и идеи..."></textarea>
</div>
</div>
</div>
</div>
```

```
<!-- Шпаргалка по textarea -->
<div class="example-section">
<h2>📋 Шпаргалка по текстовой области</h2>

<div style="display: grid; grid-template-columns: repeat(auto-fit,
minmax(300px, 1fr)); gap: 15px; margin: 20px 0;">
<div style="border: 2px solid #3498db; padding: 15px; border-radius: 8px;">
<h3 style="color: #3498db; margin-top: 0;">👉 Основные атрибуты</h3>
<ul>
<li><code>rows</code> - количество строк</li>
<li><code>cols</code> - ширина в символах</li>
<li><code>name</code> - имя для сервера</li>
<li><code>placeholder</code> - подсказка</li>
<li><code>maxlength</code> - ограничение длины</li>
</ul>
</div>
```

```
<div style="border: 2px solid #27ae60; padding: 15px; border-radius: 8px;">
<h3 style="color: #27ae60; margin-top: 0;">💡 Лучшие практики</h3>
<ul>
<li>Используйте для длинных текстов</li>
<li>Указывайте понятные подсказки</li>
<li>Ограничивайте длину при необходимости</li>
<li>Добавляйте счетчик символов</li>
<li>Тестируйте на мобильных устройствах</li>
</ul>
</div>
```

```
<div style="border: 2px solid #e74c3c; padding: 15px; border-radius: 8px;">
<h3 style="color: #e74c3c; margin-top: 0;">🚫 Чего избегать</h3>
<ul>
<li>Слишком маленьких размеров</li>
<li>Неочевидных подсказок</li>
<li>Отсутствия ограничений для критичных полей</li>
<li>Использования для коротких текстов</li>
</ul>
```

```
</div>
</div>
</div>

<!-- Практическое задание -->
<div class="example-section">
<h2> Практическое задание</h2>

<div class="practice-task">
<h3>Создайте форму "Онлайн-заявка на консультацию"</h3>

<div class="code-example">
<!DOCTYPE html>
<html>
<head>
<title>Заявка на консультацию</title>
<style>
/* Ваши стили здесь */
</style>
</head>
<body>
<h1>Онлайн-заявка на юридическую консультацию</h1>

<form action="/consultation" method="POST">
<!-- Личные данные -->
<div>
<label>ФИО:</label>
<input type="text" name="fullname" required>
</div>

<div>
<label>Телефон:</label>
<input type="tel" name="phone" required>
</div>

<!-- Ваша ЗАДАЧА: Создать текстовые области ниже -->
<!-- 1. Описание проблемы (обязательное, 10 строк) -->
<!-- 2. Дополнительная информация (необязательное, 6 строк) -->
<!-- 3. Пожелания к консультации (5 строк, с подсказкой) -->
<!-- 4. Шаблон ответа (только для чтения, 8 строк) -->
```

<!-- 5. Поле с ограничением 1000 символов и счетчиком -->

```
<input type="submit" value="📝 Отправить заявку">
</form>
</body>
</html>
</div>

<div class="note">
<strong>📝 Задание:</strong> Создайте различные текстовые области
для формы заявки:
<ul>
<li>Обязательное поле для описания проблемы</li>
<li>Необязательное поле для дополнительной информации</li>
<li>Поле с подсказкой о формате ввода</li>
<li>Текст только для чтения с шаблоном</li>
<li>Поле с ограничением символов и динамическим счетчиком</li>
<li>Разные размеры (rows) для разных целей</li>
</ul>
</div>
</div>
</div>
</body>
</html>
```

Часть 6: Что Вы теперь умеете

- Создавать базовые текстовые области
- Использовать атрибуты `rows` и `cols` для задания размеров
- Добавлять подсказки с помощью `placeholder`
- Ограничивать длину текста с `maxlength`
- Создавать поля только для чтения и отключенные
- Стилизовать текстовые области с помощью CSS
- Добавлять счетчики символов

Часть 7: Итоги и рекомендации

🎯 Ключевые принципы работы с текстовыми областями:

Используйте для многострочного текста — отзывы, сообщения, описания

Задавайте адекватные размеры — чтобы пользователь видел достаточно текста

Добавляйте понятные подсказки — помогают понять, что писать

Ограничивайте длину — защищает от переполнения базы данных

Тестируйте на разных устройствах — особенно на мобильных

 **Профессиональные советы:**

Используйте `resize: vertical` — позволяет менять только высоту
Добавляйте счетчики символов — пользователь видит, сколько осталось

Используйте моноширинный шрифт для полей с кодом

Группируйте связанные поля — используйте `<fieldset>`

Тестируйте с длинными текстами — убедитесь, что скролл работает правильно

 **Следующие шаги:**

Изучить JavaScript для динамического управления текстовыми областями

Освоить валидацию многострочного текста

Узнать о загрузке файлов вместе с текстом

Изучить Rich Text Editor для форматированного текста

Помните: Хорошо сделанная текстовая область делает процесс ввода длинных текстов комфортным и приятным для пользователя! 

11.5. Практика: создаём на странице "Гостевая книга" форму, где посетители могут оставить Вам сообщение

Введение: Ваша собственная книга отзывов в интернете

Представьте, что у Вас есть виртуальная гостиная, где друзья и родственники могут оставлять Вам сообщения, пожелания или отзывы. Гостевая книга — это именно такой инструмент! Сегодня мы создадим полноценную страницу, где посетители смогут оставлять сообщения, а Вы сможете их просматривать.

Часть 1: Планируем структуру гостевой книги

Перед тем как писать код, давайте определим, что нам нужно:

1. Форма для ввода сообщения с полями:

Имя посетителя
Email (для ответа)
Тема сообщения
Текст сообщения
Кнопка отправки

2. Область для отображения сообщений

Имя автора
Дата отправки
Тема
Текст сообщения

Часть 2: Базовая структура формы гостевой книги

```
html
<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="utf-8">
    <title>Гостевая книга - Мой сайт</title>
    <style>
        /* Стили будут добавлены позже */
    </style>
</head>
<body>
    <h1>█ Гостевая книга</h1>
    <p>Оставьте ваше сообщение, отзыв или пожелание!</p>
```

```

<!-- Форма для отправки сообщения -->
<section class="message-form">
  <h2>✍ Написать сообщение</h2>
  <form action="#" method="POST">
    <!-- Здесь будут поля формы -->
  </form>
</section>

<!-- Область с существующими сообщениями -->
<section class="messages">
  <h2>✉ Сообщения гостей</h2>
  <!-- Здесь будут отображаться сообщения -->
</section>
</body>
</html>

```

Часть 3: Создаём полноценную форму для сообщений

```

html
<form action="#" method="POST" class="guestbook-form">
  <div class="form-row">
    <div class="form-group">
      <label for="name">Ваше имя *</label>
      <input type="text" id="name" name="name" required
             placeholder="Иван Иванов">
    </div>

    <div class="form-group">
      <label for="email">Ваш email</label>
      <input type="email" id="email" name="email"
             placeholder="ivan@example.com">
    </div>
  </div>

  <div class="form-group">
    <label for="subject">Тема сообщения *</label>
    <input type="text" id="subject" name="subject" required
           placeholder="О чём Вы хотите написать?">
  </div>

  <div class="form-group">
    <label for="message">Ваше сообщение *</label>
    <textarea id="message" name="message" rows="8" required
              placeholder="Напишите ваше сообщение здесь...">
  </div>

```

Можете рассказать о впечатлениях, задать вопрос или оставить
пожелание."></textarea>

<div class="char-counter">Максимум 1000 символов. Осталось:

1000</div>

</div>

<div class="form-group">

<label class="checkbox-label">

<input type="checkbox" name="newsletter" checked>

Хочу получать уведомления об ответах

</label>

</div>

<button type="submit" class="submit-btn">

 Отправить сообщение

</button>

</form>

Часть 4: Полная страница гостевой книги с оформлением

html

<!DOCTYPE html>

<html lang="ru">

<head>

<meta charset="utf-8">

<title>Гостевая книга - Мой персональный сайт</title>

<style>

* {

margin: 0;

padding: 0;

box-sizing: border-box;

}

body {

font-family: 'Arial', sans-serif;

line-height: 1.6;

color: #333;

background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);

min-height: 100vh;

}

.container {

max-width: 800px;

margin: 0 auto;

padding: 20px;

```
}

.guestbook-page {
    background: white;
    border-radius: 15px;
    box-shadow: 0 10px 30px rgba(0,0,0,0.2);
    overflow: hidden;
}

/* Шапка страницы */
.page-header {
    background: linear-gradient(135deg, #2c3e50, #3498db);
    color: white;
    padding: 40px 30px;
    text-align: center;
}

.page-header h1 {
    font-size: 2.5em;
    margin-bottom: 10px;
}

.page-header p {
    font-size: 1.2em;
    opacity: 0.9;
}

/* Основное содержимое */
.page-content {
    padding: 30px;
}

/* Стили формы */
.message-form {
    background: #f8f9fa;
    padding: 30px;
    border-radius: 10px;
    margin-bottom: 40px;
    border: 2px solid #e9ecf;
}

.message-form h2 {
    color: #2c3e50;
    margin-bottom: 25px;
    display: flex;
```

```
    align-items: center;
    gap: 10px;
}

.form-row {
    display: grid;
    grid-template-columns: 1fr 1fr;
    gap: 20px;
    margin-bottom: 20px;
}

.form-group {
    margin-bottom: 20px;
}

label {
    display: block;
    margin-bottom: 8px;
    font-weight: bold;
    color: #2c3e50;
}

.required::after {
    content: " *";
    color: #e74c3c;
}

input[type="text"],
input[type="email"],
textarea {
    width: 100%;
    padding: 12px 15px;
    border: 2px solid #ddd;
    border-radius: 8px;
    font-size: 16px;
    transition: all 0.3s ease;
    font-family: inherit;
}

input[type="text"]:focus,
input[type="email"]:focus,
textarea:focus {
    border-color: #3498db;
    outline: none;
    box-shadow: 0 0 0 3px rgba(52, 152, 219, 0.1);
```

```
}

textarea {
    resize: vertical;
    min-height: 120px;
}

.char-counter {
    font-size: 14px;
    color: #666;
    text-align: right;
    margin-top: 5px;
}

.char-counter .warning {
    color: #e74c3c;
    font-weight: bold;
}

.checkbox-label {
    display: flex;
    align-items: center;
    gap: 10px;
    font-weight: normal;
    cursor: pointer;
}

.checkbox-label input[type="checkbox"] {
    width: auto;
}

.submit-btn {
    background: linear-gradient(135deg, #27ae60, #2ecc71);
    color: white;
    border: none;
    padding: 15px 30px;
    font-size: 18px;
    border-radius: 8px;
    cursor: pointer;
    transition: all 0.3s ease;
    display: flex;
    align-items: center;
    gap: 10px;
    margin: 0 auto;
}
```

```
.submit-btn:hover {
    transform: translateY(-2px);
    box-shadow: 0 5px 15px rgba(39, 174, 96, 0.4);
}

/* Стили для сообщений */
.messages-section {
    margin-top: 40px;
}

.messages-section h2 {
    color: #2c3e50;
    margin-bottom: 25px;
    display: flex;
    align-items: center;
    gap: 10px;
}

.message {
    background: white;
    border: 2px solid #e9ecf;
    border-radius: 10px;
    padding: 25px;
    margin-bottom: 20px;
    transition: all 0.3s ease;
}

.message:hover {
    border-color: #3498db;
    transform: translateY(-2px);
    box-shadow: 0 5px 15px rgba(0,0,0,0.1);
}

.message-header {
    display: flex;
    justify-content: space-between;
    align-items: center;
    margin-bottom: 15px;
    padding-bottom: 15px;
    border-bottom: 1px solid #eee;
}

.author-info {
    display: flex;
```

```
    align-items: center;
    gap: 15px;
}

.author-avatar {
    width: 50px;
    height: 50px;
    background: linear-gradient(135deg, #3498db, #9b59b6);
    border-radius: 50%;
    display: flex;
    align-items: center;
    justify-content: center;
    color: white;
    font-weight: bold;
    font-size: 18px;
}

.author-details h3 {
    color: #2c3e50;
    margin-bottom: 5px;
}

.author-email {
    color: #666;
    font-size: 14px;
}

.message-date {
    color: #999;
    font-size: 14px;
}

.message-subject {
    color: #2c3e50;
    font-size: 1.2em;
    font-weight: bold;
    margin-bottom: 15px;
}

.message-text {
    color: #555;
    line-height: 1.6;
    white-space: pre-line;
}
```

```
.empty-messages {
    text-align: center;
    padding: 40px;
    color: #666;
    background: #f8f9fa;
    border-radius: 10px;
    border: 2px dashed #ddd;
}

/* Адаптивность */
@media (max-width: 768px) {
    .form-row {
        grid-template-columns: 1fr;
    }
}

.message-header {
    flex-direction: column;
    align-items: flex-start;
    gap: 10px;
}

.author-info {
    width: 100%;
}
}

</style>
</head>
<body>
<div class="container">
    <div class="guestbook-page">
        <!-- Шапка страницы -->
        <header class="page-header">
            <h1>📖 Гостевая книга</h1>
            <p>Ваше мнение важно для меня! Оставляйте сообщения, отзывы и пожелания</p>
        </header>

        <!-- Основное содержимое -->
        <div class="page-content">
            <!-- Форма для отправки сообщения -->
            <section class="message-form">
                <h2>✍ Написать сообщение</h2>
                <form action="#" method="POST" class="guestbook-form">
                    <div class="form-row">
                        <div class="form-group">
```

```
<label for="name" class="required">Ваше имя</label>
<input type="text" id="name" name="name" required
placeholder="Как к Вам обращаться?">
</div>

<div class="form-group">
    <label for="email">Ваш email</label>
    <input type="email" id="email" name="email"
placeholder="Для ответа на ваше сообщение">
</div>
</div>

<div class="form-group">
    <label for="subject" class="required">Тема
сообщения</label>
    <input type="text" id="subject" name="subject" required
placeholder="О чём Вы хотите написать?">
</div>

<div class="form-group">
    <label for="message" class="required">Ваше
сообщение</label>
    <textarea id="message" name="message" rows="8"
required
placeholder="Напишите ваше сообщение здесь...">
    Можете рассказать о впечатлениях, задать вопрос или оставить
пожелание.

    Страйтесь писать понятно и по делу. Спасибо!"</textarea>
    <div class="char-counter">Максимум 1000 символов.
    Осталось: <span id="charCount">1000</span></div>
</div>

<div class="form-group">
    <label class="checkbox-label">
        <input type="checkbox" name="newsletter" checked>
 Хочу получать уведомления о ответах
    </label>
</div>

<button type="submit" class="submit-btn">
 Отправить сообщение
</button>
</form>
</section>
```

```
<!-- Существующие сообщения -->
<section class="messages-section">
    <h2>✉️ Сообщения гостей</h2>

    <div class="messages-list">
        <!-- Пример сообщения 1 -->
        <article class="message">
            <div class="message-header">
                <div class="author-info">
                    <div class="author-avatar">МИ</div>
                    <div class="author-details">
                        <h3>Мария Иванова</h3>
                        <div class="author-
email">maria@example.com</div>
                    </div>
                </div>
                <div class="message-date">15 марта 2024, 14:30</div>
            </div>

            <div class="message-subject">Благодарность за
помощь</div>
            <div class="message-text">Хочу выразить огромную
благодарность за Вашу помощь в освоении компьютера! Благодаря
Вашим урокам я теперь уверенно пользуюсь интернетом, могу общаться
с родственниками по видеосвязи и даже делать покупки онлайн.

Ваши объяснения очень понятные и доступные. Особенно понравились
уроки про электронную почту и социальные сети.

Спасибо Вам большое! Желаю успехов в Вашем важном деле!</div>
        </article>

        <!-- Пример сообщения 2 -->
        <article class="message">
            <div class="message-header">
                <div class="author-info">
                    <div class="author-avatar">ПС</div>
                    <div class="author-details">
                        <h3>Петр Сидоров</h3>
                        <div class="author-
email">petr.sidorov@mail.ru</div>
                    </div>
                </div>
                <div class="message-date">14 марта 2024, 10:15</div>
```

```
</div>

<div class="message-subject">Вопрос по таблицам в
HTML</div>
    <div class="message-text">Здравствуйте! Очень
понравился Ваш урок по созданию таблиц. Всё было понятно, пока не
дошёл до объединения ячеек.
```

Не могли бы Вы объяснить подробнее, как работает атрибут colspan?
Хочу сделать таблицу расходов за неделю, где в конце будет строка
"Итого".

Заранее спасибо за ответ!</div>
</article>

```
<!-- Пример сообщения 3 -->
<article class="message">
    <div class="message-header">
        <div class="author-info">
            <div class="author-avatar">AB</div>
            <div class="author-details">
                <h3>Анна Васильева</h3>
                <div class="author-
email">anna_vas@yandex.ru</div>
            </div>
        </div>
        <div class="message-date">13 марта 2024, 18:45</div>
    </div>

    <div class="message-subject">Поздравление с днём
рождения!</div>
    <div class="message-text">Дорогой друг!
```

Хочу поздравить Вас с днём рождения! Желаю Вам крепкого здоровья, счастья, благополучия и успехов во всех начинаниях.

Пусть Ваша работа приносит Вам радость и удовлетворение, а ученики всегда радуют своими успехами.

С наилучшими пожеланиями,
Анна</div>

```
</article>
</div>
```

```
<!-- Сообщение, когда нет записей -->
```

```
<!-- <div class="empty-messages">
    <h3>⚠ Пока нет сообщений</h3>
    <p>Будьте первым, кто оставит сообщение в гостевой
книге!</p>
</div> -->
</section>
</div>
</div>
</div>

<script>
// Счетчик символов для текстового поля
const messageTextarea = document.getElementById('message');
const charCount = document.getElementById('charCount');
const maxChars = 1000;

messageTextarea.addEventListener('input', function() {
    const remaining = maxChars - this.value.length;
    charCount.textContent = remaining;

    if (remaining < 100) {
        charCount.className = 'warning';
    } else {
        charCount.className = '';
    }

    if (remaining < 0) {
        this.value = this.value.substring(0, maxChars);
        charCount.textContent = 0;
    }
});

// Обработка отправки формы (для демонстрации)
const form = document.querySelector('.guestbook-form');
form.addEventListener('submit', function(e) {
    e.preventDefault();

    // В реальном приложении здесь был бы AJAX запрос к серверу
    alert('Сообщение отправлено! В реальном сайте оно бы
сохранилось в базе данных.\n\nВ этом учебном примере мы просто
демонстрируем работу формы.');
}

// Очистка формы после "отправки"
form.reset();
charCount.textContent = maxChars;
```

```

        charCount.className = "";
    });

// Генерация аватарок по инициалам
function generateAvatar(name) {
    const names = name.split(' ');
    let initials = "";

    if (names.length >= 2) {
        initials = names[0][0] + names[1][0];
    } else if (names.length === 1) {
        initials = names[0][0];
    }

    return initials.toUpperCase();
}

// В реальном приложении аватарки генерировались бы на сервере
// Здесь просто демонстрируем возможность
</script>
</body>
</html>

```

Часть 5: Дополнительные улучшения и функции

5.1. Валидация формы на JavaScript

```

javascript
// Добавляем в script более продвинутую валидацию
function validateForm(formData) {
    const errors = [];

    // Проверка имени
    if (formData.name.length < 2) {
        errors.push('Имя должно содержать минимум 2 символа');
    }

    // Проверка email (если указан)
    if (formData.email && !isValidEmail(formData.email)) {
        errors.push('Укажите корректный email адрес');
    }

    // Проверка темы
    if (formData.subject.length < 5) {
        errors.push('Тема сообщения слишком короткая');
    }
}

```

```

    }

    // Проверка сообщения
    if (formData.message.length < 10) {
        errors.push('Сообщение должно содержать минимум 10 символов');
    }

    return errors;
}

function isValidEmail(email) {
    const emailRegex = /^[^@\s]+@[^\s@]+\.[^\s@]+$/;
    return emailRegex.test(email);
}

```

5.2. Локальное сохранение сообщений

```

javascript
// Сохранение сообщений в localStorage (для демонстрации)
function saveMessage(message) {
    let messages = JSON.parse(localStorage.getItem('guestbookMessages')) ||
[];  

    messages.unshift({
        id: Date.now(),
        ...message,
        date: new Date().toLocaleString('ru-RU')
    });
    localStorage.setItem('guestbookMessages', JSON.stringify(messages));
    return messages;
}

function loadMessages() {
    return JSON.parse(localStorage.getItem('guestbookMessages')) || [];
}

```

Часть 6: Что Вы создали в этом уроке

- Полноценную страницу гостевой книги с современным дизайном
- Интуитивно понятную форму для отправки сообщений
- Валидацию полей на стороне клиента
- Счетчик символов для текстового поля
- Красивое отображение существующих сообщений
- Адаптивный дизайн для мобильных устройств
- Интерактивные элементы с анимациями

Часть 7: Итоги и рекомендации по развитию

Ключевые элементы гостевой книги:

1. Простая форма — минимум обязательных полей
2. Валидация данных — предотвращает ошибки
3. Привлекательный дизайн — побуждает к участию
4. Удобное отображение — сообщения легко читать
5. Обратная связь — пользователь видит результат

Идеи для улучшения:

Добавьте рейтинг — звёзды для оценки
Реализуйте ответы — возможность отвечать на сообщения
Добавьте модерацию — предпросмотр перед публикацией
Внедрите капчу — защита от спама
Добавьте сортировку — по дате, рейтингу, автору

Следующие шаги для реального проекта:

1. Настройте серверную часть (PHP, Node.js, Python)
2. Подключите базу данных для хранения сообщений
3. Реализуйте панель администратора для управления
4. Добавьте email уведомления о новых сообщениях
5. Настройте защиту от спама и вредоносных данных

Помните: Гостевая книга — это не просто форма, а способ общения с Вашими посетителями. Сделайте этот процесс приятным и удобным! 

Ваша готовая гостевая книга теперь может принимать сообщения от друзей, родственников и всех, кто посещает Ваш сайт! 

Глава 12: Проверяем свою работу и исправляем ошибки

12.1. Что такое "валидатор" и зачем он нужен? (Простая аналогия с проверкой орфографии)

Введение: Ваш личный корректор для кода

Представьте, что Вы написали важное письмо и хотите быть уверены, что в нём нет ошибок. Вы просите друга проверить орфографию и пунктуацию. Валидатор в веб-разработке — это такой же "друг-корректор", но для Вашего HTML-кода! Он проверяет, всё ли Вы сделали правильно и не нарушили ли правила языка.

Часть 1: Простая аналогия — проверка орфографии в Word

Давайте проведём параллель с тем, что знакомо каждому:

```
html
<!-- Аналогия: Документ Word с ошибками -->
<h1>Привет, мир!<h1>
<!-- Ошибка: не закрыт тег -->
<p>Это мой website. <!-- Ошибка: английское слово в русском тексте -->
 <!-- Ошибка: нет атрибута alt -->
```

Word подчеркнёт красным: "website" (предложит исправить на "веб-сайт")

Валидатор подчеркнёт: незакрытый тег и отсутствие alt

Часть 2: Что такое валидатор на самом деле?

Валидатор — это специальная программа или онлайн-сервис, который проверяет HTML-код на соответствие стандартам W3C (World Wide Web Consortium).

```
html
<!-- Пример кода с ошибками, которые найдет валидатор -->

<!-- ОШИБКА 1: Незакрытый тег -->
<div>
  <p>Текст
</div>

<!-- ОШИБКА 2: Неправильная вложенность -->
<p><strong><em>Текст</em></strong></p>
```

```
<!-- ОШИБКА 3: Отсутствует обязательный атрибут -->

```

```
<!-- ОШИБКА 4: Устаревший тег -->
<font size="5">Большой текст</font>
```

Часть 3: Типы валидаторов и где их найти

3.1. Онлайн-валидаторы

Самый популярный — [Validator.w3.org](https://validator.w3.org):

```
html
<!-- Пример проверки на validator.w3.org -->
<!DOCTYPE html>
<html>
<head>
  <title>Страница с ошибками</title>
</head>
<body>
  <h1>Заголовок <!-- ОШИБКА: нет закрывающего тега -->
   <!-- ПРЕДУПРЕЖДЕНИЕ: нет alt -->
</body>
</html>
```

Результат проверки:

✖ Error: End tag for h1 missing
⚠ Warning: Consider adding alt attribute

3.2. Валидаторы в редакторах кода

```
html
<!-- VS Code, Sublime Text, WebStorm показывают ошибки сразу -->
<div class="container">
  <p>Текст внутри</p>
</div>
<!-- Редактор подсветит: ожидается закрывающая скобка -->
```

Часть 4: Полная страница с объяснением и примерами

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

```
<meta charset="utf-8">
<title>Валидаторы в HTML: Ваш код-корректор</title>
<style>
  body {
    font-family: 'Arial', sans-serif;
    max-width: 1000px;
    margin: 0 auto;
    padding: 20px;
    line-height: 1.6;
    background-color: #f8f9fa;
  }
  .example-section {
    background: white;
    padding: 25px;
    margin: 25px 0;
    border-radius: 10px;
    box-shadow: 0 2px 5px rgba(0,0,0,0.1);
  }
  h1, h2 {
    color: #2c3e50;
  }
  h1 {
    border-bottom: 3px solid #3498db;
    padding-bottom: 15px;
  }
  .code-example {
    background: #2d2d2d;
    color: #f8f8f2;
    padding: 15px;
    border-radius: 5px;
    font-family: 'Courier New', monospace;
    margin: 15px 0;
    overflow-x: auto;
    position: relative;
  }
  .error-code {
    background: #ffebee;
    border-left: 4px solid #e74c3c;
    padding: 15px;
    margin: 15px 0;
    border-radius: 5px;
  }
  .fixed-code {
    background: #e8f5e8;
    border-left: 4px solid #27ae60;
    padding: 15px;
    margin: 15px 0;
    border-radius: 5px;
  }
</style>
```

```
padding: 15px;
margin: 15px 0;
border-radius: 5px;
}
.note {
background: #e7f3ff;
padding: 15px;
border-left: 4px solid #2196F3;
margin: 15px 0;
border-radius: 5px;
}
.warning {
background: #fff3e0;
padding: 15px;
border-left: 4px solid #ff9800;
margin: 15px 0;
border-radius: 5px;
}
.validator-demo {
background: #f5f5f5;
border: 2px solid #ddd;
border-radius: 8px;
padding: 20px;
margin: 20px 0;
}
.error-marker {
background: #ffcdd2;
color: #c62828;
padding: 2px 4px;
border-radius: 3px;
font-weight: bold;
}
.success-marker {
background: #c8e6c9;
color: #2e7d32;
padding: 2px 4px;
border-radius: 3px;
font-weight: bold;
}
.comparison-table {
width: 100%;
border-collapse: collapse;
margin: 20px 0;
}
.comparison-table th, .comparison-table td {
```

```
border: 1px solid #ddd;
padding: 12px;
text-align: left;
}
.comparison-table th {
background: #34495e;
color: white;
}
.comparison-table tr:nth-child(even) {
background: #f9f9f9;
}
</style>
</head>
<body>
<h1> Валидатор HTML: Ваш личный инспектор кода</h1>

<div class="note">
<strong> Простая аналогия:</strong> Валидатор — это как проверка орфографии в Word, но для HTML-кода. Он находит ошибки, указывает на проблемы и помогает писать корректный, стандартный код.
</div>

<!-- Аналогия с Word -->
<div class="example-section">
<h2>1. Аналогия: Проверка орфографии в Word vs Валидация HTML</h2>

<table class="comparison-table">
<tr>
<th>Microsoft Word</th>
<th>HTML Валидатор</th>
<th>Объяснение</th>
</tr>
<tr>
<td>Подчёркивает орфографические ошибки красным</td>
<td>Показывает синтаксические ошибки в коде</td>
<td>Оба находят нарушения правил языка</td>
</tr>
<tr>
<td>Предлагает варианты исправления</td>
<td>Указывает, как исправить ошибку</td>
<td>Помогают улучшить качество текста/кода</td>
</tr>
<tr>
<td>Проверяет пунктуацию</td>
```

```
<td>Проверяет правильность структуры тегов</td>
<td>Следят за правильным оформлением</td>
</tr>
<tr>
    <td>Можно игнорировать ошибки (но не стоит)</td>
    <td>Можно пропустить ошибки (но страдает качество)</td>
    <td>Ошибки лучше исправлять сразу</td>
</tr>
</table>

<div class="validator-demo">
    <h3>📝 Пример из жизни:</h3>
    <p><strong>В Word Вы написали:</strong> "Сегодня я иду в
кина."</p>
    <p><strong>Word подчеркнёт:</strong> "кина" и предложит:
"кино"</p>

    <p><strong>В HTML Вы написали:</strong></p>
    <div class="code-example">
<div>
    <p>Текст абзаца
</div>
    </div>
    <p><strong>Валидатор покажет:</strong> "End tag p missing" (Не
хватает закрывающего тега p)</p>
    </div>
</div>

<!-- Что такое валидатор -->
<div class="example-section">
    <h2>2. Что такое валидатор и как он работает?</h2>

    <div class="note">
        <strong>🎯 Определение:</strong> Валидатор — это программа,
которая проверяет HTML-код на соответствие официальным стандартам
W3C (Консорциум Всемирной паутины).
    </div>

    <div class="code-example">
<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="utf-8">
    <title>Пример страницы для валидации</title>
</head>
```

```
<body>
  <h1>Заголовок страницы</h1>
  <p>Абзац текста с <strong>выделением</strong>. </p>
  
</body>
</html>
</div>

<p><strong>Валидатор проверяет:</strong></p>
<ul>
  <li>✓ Правильность структуры тегов</li>
  <li>✓ Корректность вложенности элементов</li>
  <li>✓ Наличие обязательных атрибутов</li>
  <li>✓ Соответствие объявленному типу документа
(DOCTYPE)</li>
  <li>✗ Использование устаревших тегов и атрибутов</li>
</ul>
</div>
```

```
<!-- Примеры ошибок -->
<div class="example-section">
  <h2>3. Примеры ошибок, которые находит валидатор</h2>
```

```
<div class="error-code">
  <h3>✗ Код с ошибками:</h3>
  <div class="code-example">
<!DOCTYPE html>
<html>
<head>
  <title>Страница с ошибками</title>
</head>
<body>
  <h1>Заголовок первого уровня <!-- ОШИБКА: нет закрывающего тега
-->
  <p>Абзац с <strong><em>выделением</strong></em> <!-- ОШИБКА:
неправильная вложенность -->
     <!-- ПРЕДУПРЕЖДЕНИЕ: нет атрибута alt -->
    <center>Центрированный текст</center> <!-- ОШИБКА: устаревший тег
-->
    <div>
      <p>Ещё один абзац
    </div>
</body>
</html>
</div>
```

```
</div>

<div class="fixed-code">
    <h3>✓ Исправленный код:</h3>
    <div class="code-example">
<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="utf-8">
    <title>Страница с ошибками</title>
</head>
<body>
    <h1>Заголовок первого уровня</h1>
    <p>Абзац с <strong><em>выделением</em></strong></p>
    
    <div style="text-align: center;">
        <p>Центрированный текст</p>
    </div>
    <div>
        <p>Ещё один абзац</p>
    </div>
</body>
</html>
    </div>
    </div>
</div>

<!-- Зачем нужен валидатор -->
<div class="example-section">
    <h2>4. Зачем вообще нужен валидатор? 6 главных причин</h2>

    <div style="display: grid; grid-template-columns: repeat(auto-fit, minmax(300px, 1fr)); gap: 15px; margin: 20px 0;">
        <div style="border: 2px solid #3498db; padding: 15px; border-radius: 8px;">
            <h3 style="color: #3498db; margin-top: 0;">🌐 1. Кроссбраузерность</h3>
            <p>Валидный код одинаково хорошо работает во всех браузерах (Chrome, Firefox, Safari, Edge)</p>
        </div>

        <div style="border: 2px solid #27ae60; padding: 15px; border-radius: 8px;">
            <h3 style="color: #27ae60; margin-top: 0;">♿ 2. Доступность</h3>
```

<p>Правильный HTML лучше работает с программами чтения для слабовидящих</p>

</div>

<div style="border: 2px solid #e74c3c; padding: 15px; border-radius: 8px;">

 <h3 style="color: #e74c3c; margin-top: 0;"> 3. Скорость загрузки</h3>

 <p>Браузеры быстрее обрабатывают валидный код</p>

</div>

<div style="border: 2px solid #f39c12; padding: 15px; border-radius: 8px;">

 <h3 style="color: #f39c12; margin-top: 0;"> 4. Адаптивность</h3>

 <p>Корректный HTML лучше адаптируется под мобильные устройства</p>

</div>

<div style="border: 2px solid #9b59b6; padding: 15px; border-radius: 8px;">

 <h3 style="color: #9b59b6; margin-top: 0;"> 5. SEO-оптимизация</h3>

 <p>Поисковые системы лучше индексируют валидные страницы</p>

</div>

<div style="border: 2px solid #1abc9c; padding: 15px; border-radius: 8px;">

 <h3 style="color: #1abc9c; margin-top: 0;"> 6. Простота поддержки</h3>

 <p>Код без ошибок легче понимать и изменять</p>

</div>

</div>

</div>

<!-- Как пользоваться валидатором -->

<div class="example-section">

 <h2>5. Как пользоваться онлайн-валидатором W3C</h2>

<div class="note">

  Практическое руководство:

</div>

<ol style="margin: 20px 0; padding-left: 20px;">

```
<li><strong>Перейдите на сайт:</strong> <a href="https://validator.w3.org" target="_blank">validator.w3.org</a></li>
<li><strong>Выберите способ проверки:</strong>
<ul>
    <li>По URL (если сайт уже в интернете)</li>
    <li>Загрузка файла (если есть HTML-файл)</li>
    <li>Ввод кода (для быстрой проверки фрагмента)</li>
</ul>
</li>
<li><strong>Нажмите "Check":</strong> Валидатор проанализирует код</li>
<li><strong>Изучите результаты:</strong> Ошибки (красным) и предупреждения (жёлтым)</li>
<li><strong>Исправьте ошибки:</strong> Следуйте рекомендациям валидатора</li>
</ol>
```

```
<div class="validator-demo">
<h3>📋 Пример отчёта валидатора:</h3>
<div style="background: white; padding: 15px; border-radius: 5px; font-family: monospace;">
    <div style="color: #e74c3c;">✖ Error: End tag for h1 missing</div>
    <div style="margin-left: 20px;">Line 8, Column 1: </body></div>
    <div style="color: #f39c12;">⚠ Warning: Consider adding alt attribute</div>
    <div style="margin-left: 20px;">Line 7, Column 1: </div>
    </div>
</div>
</div>
```

```
<!-- Частые ошибки новичков -->
<div class="example-section">
<h2>6. Типичные ошибки, которые находит валидатор</h2>
```

Ошибка	Пример	Исправление	Почему важно
Незакрытые теги	<code><p>Текст</code></td>		

```
<td><code><p>Текст</p></code></td>
<td>Браузер не понимает, где заканчивается элемент</td>
</tr>
<tr>
<td>Неправильная вложенность</td>
<td><code><p><strong>Текст</p></strong></code></td>
<td><code><p><strong>Текст</strong></p></code></td>
<td>Нарушает древовидную структуру документа</td>
</tr>
<tr>
<td>Отсутствие alt у изображений</td>
<td><code></code></td>
<td><code></code></td>
<td>Важно для доступности и SEO</td>
</tr>
<tr>
<td>Устаревшие теги</td>
<td><code><center><font></font></center></code></td>
<td>Использовать CSS</td>
<td>Современные браузеры могут не поддерживать</td>
</tr>
<tr>
<td>Отсутствие doctype</td>
<td>Нет <code><!DOCTYPE html></code></td>
<td>Добавить в начало</td>
<td>Браузер не понимает версию HTML</td>
</tr>
</table>
</div>
```

```
<!-- Практическое задание -->
<div class="example-section">
<h2>🎯 Практическое задание: Найди ошибки!</h2>

<div class="practice-task">
<h3>Проверь свои навыки: найди 5 ошибок в этом коде</h3>

<div class="code-example">
<!DOCTYPE html>
<html>
<head>
<title>Мой сайт</title>
</head>
<body>
<h1>Добро пожаловать на мой сайт
```

```
<p>Это мой первый <strong><em>веб-сайт</strong></em>.<br><br><br><center><font size="5">Большой текст</font></center><div><p>Контент внутри дива</p></div></body></html></div>

<div class="note">
    <strong>📝 Задание:</strong> Найдите все ошибки и напишите исправленный вариант кода.
    Затем проверьте его на <a href="https://validator.w3.org" target="_blank">validator.w3.org</a>!
</div>

<details style="margin-top: 15px;">
    <summary><strong>💡 Подсказка: Какие ошибки искать?</strong></summary>
        <ul>
            <li>Незакрытые теги</li>
            <li>Неправильная вложенность</li>
            <li>Отсутствие важных атрибутов</li>
            <li>Устаревшие теги</li>
            <li>Отсутствие атрибута lang</li>
        </ul>
    </details>
</div>
</div>

<!-- Шпаргалка --&gt;
&lt;div class="example-section"&gt;
    &lt;h2&gt;📝 Шпаргалка по валидации&lt;/h2&gt;

        &lt;div style="display: grid; grid-template-columns: repeat(auto-fit, minmax(250px, 1fr)); gap: 15px; margin: 20px 0;"&gt;
            &lt;div style="border: 2px solid #3498db; padding: 15px; border-radius: 8px;"&gt;
                &lt;h3 style="color: #3498db; margin-top: 0;"&gt;✓ Обязательные элементы&lt;/h3&gt;
                &lt;ul&gt;</pre>
```

```

<li><code><!DOCTYPE html></code></li>
<li><code><html lang="ru"></code></li>
<li><code><meta charset="utf-8"></code></li>
<li><code><title></code></li>
</ul>
</div>

<div style="border: 2px solid #e74c3c; padding: 15px; border-radius: 8px;">
    <h3 style="color: #e74c3c; margin-top: 0;">🚫 Частые ошибки</h3>
    <ul>
        <li>Незакрытые теги</li>
        <li>Нет alt у img</li>
        <li>Неправильная вложенность</li>
        <li>Устаревшие теги</li>
    </ul>
</div>

<div style="border: 2px solid #27ae60; padding: 15px; border-radius: 8px;">
    <h3 style="color: #27ae60; margin-top: 0;">🔧 Инструменты</h3>
    <ul>
        <li>validator.w3.org</li>
        <li>Встроенные в редакторы кода</li>
        <li>Браузерные расширения</li>
        <li>Инструменты разработчика</li>
    </ul>
</div>
</div>
</body>
</html>

```

Часть 7: Что Вы теперь знаете о валидаторах

- Понимаете аналогию с проверкой орфографии в Word
- Знаете, что такое валидатор и как он работает
- Можете находить типичные ошибки в HTML-коде
- Понимаете важность валидации для качества сайта
- Умеете пользоваться онлайн-валидатором W3C
- Знаете частые ошибки новичков и как их избежать

Часть 8: Итоги и рекомендации

 **Ключевые выводы:**

- 1. Валидатор — это помощник, а не критик**
- 2. Проверять код нужно регулярно, а не только в конце**
- 3. Валидный код — это профессиональный код**
- 4. Ошибки лучше находить сразу, чем исправлять последствия**

 **Практические советы:**

- Проверяйте каждую страницу перед публикацией**
- Исправляйте ошибки по порядку** — иногда одна ошибка вызывает несколько сообщений
- Не игнорируйте предупреждения** — они указывают на потенциальные проблемы
- Используйте валидацию как учебный инструмент** — учитесь на ошибках

 **Следующие шаги:**

- Начните использовать валидатор для своих проектов**
- Изучите инструменты валидации в Вашем редакторе кода**
- Узнайте о валидации CSS и доступности (accessibility)**
- Изучите семантическую вёрстку для ещё лучшего качества кода**

Помните: Валидатор — это Ваш друг, который помогает стать лучше как разработчик. Используйте его регулярно, и Ваш код будет чистым, профессиональным и надёжным! 

12.2. Простой онлайн-инструмент для проверки HTML-кода

Введение: Ваша виртуальная мастерская по ремонту кода

Представьте, что у Вас есть автомобиль, и Вы хотите быть уверены, что всё работает исправно. Вы едете в автосервис, где мастер проводит диагностику. Онлайн-валидатор — это такой же "автосервис", но для Вашего HTML-кода! Он проводит полную диагностику и показывает, что нужно починить.

Часть 1: Самый популярный инструмент — Validator.w3.org

W3C Markup Validation Service — это официальный валидатор от организации, которая создаёт стандарты для интернета.

```
html
<!-- Пример кода для проверки в валидаторе -->
<!DOCTYPE html>
<html lang="ru">
<head>
  <meta charset="utf-8">
  <title>Мой сайт</title>
</head>
<body>
  <h1>Привет, мир!</h1>
  <p>Это мой первый сайт</p>
  
</body>
</html>
```

Часть 2: Пошаговая инструкция использования W3C Validator

Шаг 1: Открываем валидатор

Переходим на: <https://validator.w3.org>

Мы увидим три способа проверки:

text

- [Validate by URI] - Проверка по адресу сайта
- [Validate by File Upload] - Проверка загруженного файла
- [Validate by Direct Input] - Проверка введенного кода

Шаг 2: Выбираем способ проверки

Способ 1: Проверка по адресу (URI)

```
html
<!-- Если сайт уже в интернете -->
Вставляем: https://мойсайт.ru
Нажимаем: CHECK
```

Способ 2: Загрузка файла

```
html
<!-- Если есть HTML-файл на компьютере -->
1. Нажимаем "Choose File"
2. Выбираем файл index.html
3. Нажимаем "Check"
```

Способ 3: Прямой ввод кода

```
html
<!-- Для быстрой проверки фрагмента кода -->
1. Вставляем код в текстовое поле
2. Нажимаем "Check"
```

Часть 3: Полная страница с практическими примерами

```
html
<!DOCTYPE html>
<html lang="ru">
<head>
  <meta charset="utf-8">
  <title>Онлайн-валидаторы HTML: Простые инструменты для проверки
кода</title>
  <style>
    body {
      font-family: 'Arial', sans-serif;
      max-width: 1200px;
      margin: 0 auto;
      padding: 20px;
      line-height: 1.6;
      background-color: #f8f9fa;
    }
    .example-section {
      background: white;
      padding: 25px;
      margin: 25px 0;
      border-radius: 10px;
      box-shadow: 0 2px 5px rgba(0,0,0,0.1);
    }
  </style>
</head>
<body>
  <div class="example-section">
    <p>Пример HTML-кода для проверки:</p>
    <pre><!-- Код здесь --></pre>
  </div>
</body>
</html>
```

```
}

h1, h2 {
    color: #2c3e50;
}

h1 {
    border-bottom: 3px solid #3498db;
    padding-bottom: 15px;
}

.code-example {
    background: #2d2d2d;
    color: #f8f8f2;
    padding: 15px;
    border-radius: 5px;
    font-family: 'Courier New', monospace;
    margin: 15px 0;
    overflow-x: auto;
}

.note {
    background: #e7f3ff;
    padding: 15px;
    border-left: 4px solid #2196F3;
    margin: 15px 0;
    border-radius: 5px;
}

.warning {
    background: #fff3e0;
    padding: 15px;
    border-left: 4px solid #ff9800;
    margin: 15px 0;
    border-radius: 5px;
}

.success {
    background: #e8f5e8;
    padding: 15px;
    border-left: 4px solid #4CAF50;
    margin: 15px 0;
    border-radius: 5px;
}

.validator-demo {
    background: #f5f5f5;
    border: 2px solid #ddd;
    border-radius: 8px;
    padding: 20px;
    margin: 20px 0;
}
```

```
.step-by-step {
  background: white;
  border: 1px solid #e0e0e0;
  border-radius: 8px;
  padding: 0;
  margin: 20px 0;
  overflow: hidden;
}

.step {
  padding: 20px;
  border-bottom: 1px solid #e0e0e0;
}

.step:last-child {
  border-bottom: none;
}

.step-number {
  background: #3498db;
  color: white;
  width: 30px;
  height: 30px;
  border-radius: 50%;
  display: inline-flex;
  align-items: center;
  justify-content: center;
  margin-right: 10px;
  font-weight: bold;
}

.tool-comparison {
  display: grid;
  grid-template-columns: repeat(auto-fit, minmax(300px, 1fr));
  gap: 20px;
  margin: 20px 0;
}

.tool-card {
  border: 2px solid #3498db;
  border-radius: 10px;
  padding: 20px;
  background: white;
}

.tool-card h3 {
  color: #3498db;
  margin-top: 0;
  display: flex;
  align-items: center;
  gap: 10px;
```

```
        }
    .screenshot {
        background: #2c3e50;
        color: white;
        padding: 15px;
        border-radius: 5px;
        margin: 15px 0;
        font-family: 'Courier New', monospace;
    }
    .error-example {
        background: #ffebee;
        border-left: 4px solid #e74c3c;
        padding: 15px;
        margin: 15px 0;
        border-radius: 5px;
    }
    .browser-demo {
        background: #e3f2fd;
        border: 1px solid #2196F3;
        border-radius: 8px;
        padding: 15px;
        margin: 15px 0;
    }
}
</style>
</head>
<body>
    <h1>🛠 Простые онлайн-инструменты для проверки HTML-кода</h1>

    <div class="note">
        <strong>💡 Основная идея:</strong> Онлайн-валидаторы — это
        бесплатные инструменты, которые работают прямо в браузере.
        Вам не нужно ничего устанавливать на компьютер — просто
        заходите на сайт, вставляете код и получаете результат!
    </div>

    <!-- W3C Validator -->
    <div class="example-section">
        <h2>1. W3C Markup Validation Service — Официальный
        стандарт</h2>

        <div class="note">
            <strong>🌐 Адрес:</strong> <a href="https://validator.w3.org"
target="_blank">validator.w3.org</a><br>
            <strong>🎯 Назначение:</strong> Проверка HTML-кода на
            соответствие стандартам W3C
        </div>
    </div>

```

```
</div>

<div class="step-by-step">
  <div class="step">
    <div class="step-number">1</div>
    <strong>Открываем валидатор</strong>
    <p>Переходим на <a href="https://validator.w3.org"
target="_blank">validator.w3.org</a></p>
    <div class="screenshot">
      > validator.w3.org<br>
      > =====<br>
      > [ Validate by URI ]<br>
      > [ Validate by File Upload ]<br>
      > [ Validate by Direct Input ]
        </div>
    </div>

    <div class="step">
      <div class="step-number">2</div>
      <strong>Выбираем способ проверки</strong>

      <div class="browser-demo">
        <h4>📝 Validate by Direct Input (самый простой способ)</h4>
        <p>Подходит для быстрой проверки фрагментов кода</p>
        <div class="code-example">
          <!-- Вставляем свой код сюда -->
<!DOCTYPE html>
<html>
<head>
  <title>Мой сайт</title>
</head>
<body>
  <h1>Привет!</h1>
</body>
</html>
        </div>
      </div>
    </div>

    <div class="step">
      <div class="step-number">3</div>
      <strong>Нажимаем "Check" и получаем результат</strong>

      <div class="success">
        <h4>✅ Успешная проверка:</h4>
      </div>
    </div>
  </div>
</div>
```

```
<div class="screenshot">
```

✓ Document checking completed. No errors or warnings to show.

```
    </div>
```

```
</div>
```

```
<div class="error-example">
```

```
    <h4>✖ Обнаружены ошибки:</h4>
```

```
    <div class="screenshot">
```

Error: End tag for h1 missing

Line 7, Column 1: </body>

✗ Validation failed: 1 error

```
        </div>
```

```
    </div>
```

```
    </div>
```

```
    </div>
```

```
</div>
```

```
<!-- Другие инструменты -->
```

```
<div class="example-section">
```

<h2>2. Другие популярные онлайн-валидаторы</h2>

```
<div class="tool-comparison">
```

```
    <div class="tool-card">
```

```
        <h3>🔍 HTML Validator (FreeFormatter.com)</h3>
```

<p>Адрес: freeformatter.com/html-validator.html</p>

<p>Преимущества:</p>

```
<ul>
```

```
    <li>Более понятный интерфейс</li>
```

```
    <li>Подсветка ошибок в коде</li>
```

```
    <li>Подробные объяснения</li>
```

```
</ul>
```

```
    <div class="code-example">
```

```
<!-- Пример использования: -->
```

1. Вставляем код

2. Нажимаем "Validate"

3. Видим ошибки прямо в коде

```
        </div>
```

```
</div>
```

```
<div class="tool-card">
```

```
    <h3>📋 CSS HTML Validator</h3>
```

<p>Адрес: htmlvalidator.com</p>

<p>Преимущества:</p>

```
<ul>
```

```
<li>Проверка и HTML, и CSS</li>
<li>Проверка доступности</li>
<li>SEO-анализ</li>
</ul>
<div class="code-example">
<!-- Функционал: -->
✓ Проверка HTML/CSS
✓ Доступность (accessibility)
✓ SEO-оптимизация
✓ Скорость загрузки
</div>
</div>

<div class="tool-card">
<h3> Validator.nu</h3>
<p><strong>Адрес:</strong> validator.nu</p>
<p><strong>Преимущества:</strong></p>
<ul>
<li>Современные стандарты HTML5</li>
<li>Быстрая проверка</li>
<li>Поддержка новых тегов</li>
</ul>
<div class="code-example">
<!-- Особенности: -->
• HTML5 валидация
• Быстрая работа
• Современные стандарты
</div>
</div>
</div>
</div>

<!-- Практический пример -->
<div class="example-section">
<h2>3. Практический пример: Проверяем реальный код</h2>

<div class="validator-demo">
<h3> Проверим этот код в онлайн-валидаторе:</h3>

<div class="code-example">
<!DOCTYPE html>
<html>
<head>
<title>Мой учебный сайт</title>
</head>
```

```
<body>
  <h1>Добро пожаловать
  <p>Это мой первый сайт с <strong><em>выделением</strong></em>
  
  <center>Центрированный текст</center>
</body>
</html>
  </div>

  <div class="note">
    <strong>🎯 Задание:</strong>
    <ol>
      <li>Скопируйте код выше</li>
      <li>Перейдите на <a href="https://validator.w3.org"
target=_blank>validator.w3.org</a></li>
        <li>Выберите "Validate by Direct Input"</li>
        <li>Вставьте код и нажмите "Check"</li>
        <li>Запишите все найденные ошибки</li>
    </ol>
    </div>
  </div>

  <details>
    <summary><strong>💡 Ожидаемый результат
проверки:</strong></summary>
    <div class="error-example">
      <h4>✖ Валидатор покажет эти ошибки:</h4>
      <ul>
        <li><strong>Error:</strong> End tag for h1 missing</li>
        <li><strong>Error:</strong> Element em not allowed as child of
element strong in this context</li>
        <li><strong>Warning:</strong> Consider adding alt attribute to
img</li>
        <li><strong>Error:</strong> Element center not allowed as child
of element body in this context</li>
      </ul>
    </div>
  </details>
</div>

<!-- Инструменты для разных задач --&gt;
&lt;div class="example-section"&gt;
  &lt;h2&gt;4. Специализированные инструменты для разных задач&lt;/h2&gt;</pre>
```

```
<div style="display: grid; grid-template-columns: repeat(auto-fit, minmax(250px, 1fr)); gap: 15px; margin: 20px 0;">
    <div style="border: 2px solid #3498db; padding: 15px; border-radius: 8px;">
        <h3 style="color: #3498db; margin-top: 0;"> Мобильная верстка</h3>
        <p><strong>Google Mobile-Friendly Test</strong></p>
        <p>search.google.com/test/mobile-friendly</p>
        <p>Проверяет адаптивность для смартфонов</p>
    </div>

    <div style="border: 2px solid #27ae60; padding: 15px; border-radius: 8px;">
        <h3 style="color: #27ae60; margin-top: 0;"> Доступность</h3>
        <p><strong>WAVE Web Accessibility Evaluation Tool</strong></p>
        <p>wave.webaim.org</p>
        <p>Проверяет доступность для людей с ограничениями</p>
    </div>

    <div style="border: 2px solid #e74c3c; padding: 15px; border-radius: 8px;">
        <h3 style="color: #e74c3c; margin-top: 0;"> Скорость</h3>
        <p><strong>Google PageSpeed Insights</strong></p>
        <p>developers.google.com/speed/pagespeed/insights</p>
        <p>Анализирует скорость загрузки страницы</p>
    </div>

    <div style="border: 2px solid #f39c12; padding: 15px; border-radius: 8px;">
        <h3 style="color: #f39c12; margin-top: 0;"> SEO</h3>
        <p><strong>SEO Site Checkup</strong></p>
        <p>seositecheckup.com</p>
        <p>Проверяет SEO-оптимизацию страницы</p>
    </div>
</div>

<!-- Браузерные инструменты -->
<div class="example-section">
    <h2>5. Встроенные инструменты в браузерах</h2>

    <div class="note">
        <strong> Не нужно даже открывать другие сайты!</strong>
    <div>
```

Современные браузеры имеют встроенные инструменты для проверки кода.

```
</div>

<div class="step-by-step">
  <div class="step">
    <div class="step-number">1</div>
    <strong>Открываем Инструменты разработчика</strong>
    <p><strong>Windows/Linux:</strong> F12 или Ctrl+Shift+I</p>
    <p><strong>Mac:</strong> Cmd+Option+I</p>
    <p><strong>Простой способ:</strong> Правой кнопкой →
    "Просмотреть код"</p>
  </div>

  <div class="step">
    <div class="step-number">2</div>
    <strong>Находим вкладку "Console" (Консоль)</strong>
    <p>Здесь браузер показывает ошибки и предупреждения в
    реальном времени</p>
    <div class="screenshot">
      [Console Tab]<br>
      > Error: Uncaught SyntaxError: ...
      > Warning: ...
      > Info: ...
    </div>
  </div>

  <div class="step">
    <div class="step-number">3</div>
    <strong>Используем вкладку "Elements" (Элементы)</strong>
    <p>Показывает HTML-структуру страницы и подсвечивает
    проблемы</p>
    <div class="code-example">
      <!-- Подсвечивает: -->
      • Невалидные атрибуты
      • Неправильную вложенность
      • Отсутствующие элементы
    </div>
  </div>
</div>

<!-- Практическое задание -->
<div class="example-section">
  <h2>🎯 Практическое задание: Стань детективом кода!</h2>

  <div class="practice-task">
```

<h3>Найди и исправь все ошибки в этом коде</h3>

```
<div class="code-example">
<!DOCTYPE html>
<html>
<head>
    <title>Мой блог о садоводстве</title>
<body>
    <h1>Мой сад
    <p>В моем саду растут <strong><em>помидоры, огурцы и
цветы</strong></em>

    <table border="1">
        <tr>
            <td>Овощ</td>
            <td>Количество
        </tr>
        <tr>
            <td>Помидоры</td>
            <td>10 кустов
        </tr>
    </table>

    <font size="4">Приходите в гости!</font>
</body>
</html>
</div>

<div class="note">
    <strong>📝 Задание:</strong>
    <ol>
        <li>Скопируйте код выше</li>
        <li>Проверьте его в <a href="https://validator.w3.org"
target="_blank">W3C Validator</a></li>
        <li>Запишите все найденные ошибки</li>
        <li>Исправьте код</li>
        <li>Проверьте исправленный код</li>
        <li>Добейтесь сообщения "No errors or warnings"</li>
    </ol>
</div>

<details style="margin-top: 15px;">
```

```
<summary><strong>💡 Подсказка: Какие ошибки искать?  
</strong></summary>  
    <ul>  
        <li>Незакрытые теги</li>  
        <li>Неправильная вложенность</li>  
        <li>Отсутствие обязательных элементов</li>  
        <li>Устаревшие теги</li>  
        <li>Пропущенные атрибуты</li>  
    </ul>  
    </details>  
</div>  
</div>
```

<!-- Шпаргалка по инструментам -->

```
<div class="example-section">  
    <h2>📋 Шпаргалка по онлайн-инструментам</h2>  
  
    <div style="display: grid; grid-template-columns: repeat(auto-fit, minmax(300px, 1fr)); gap: 15px; margin: 20px 0;">  
        <div style="border: 2px solid #3498db; padding: 15px; border-radius: 8px;">  
            <h3 style="color: #3498db; margin-top: 0;">🔧 Основные  
            валидаторы</h3>  
            <ul>  
                <li><strong>W3C Validator</strong> - validator.w3.org</li>  
                <li><strong>FreeFormatter</strong> - freeformatter.com</li>  
                <li><strong>HTML Validator</strong> - htmlvalidator.com</li>  
                <li><strong>Validator.nu</strong> - validator.nu</li>  
            </ul>  
        </div>  
  
        <div style="border: 2px solid #27ae60; padding: 15px; border-radius: 8px;">  
            <h3 style="color: #27ae60; margin-top: 0;">🎯  
            Специализированные инструменты</h3>  
            <ul>  
                <li><strong>Mobile Test</strong> - Google Mobile-Friendly</li>  
                <li><strong>Accessibility</strong> - WAVE</li>  
                <li><strong>Speed Test</strong> - PageSpeed Insights</li>  
                <li><strong>SEO Check</strong> - SEO Site Checkup</li>  
            </ul>  
        </div>  
  
        <div style="border: 2px solid #e74c3c; padding: 15px; border-radius: 8px;">
```

```
<h3 style="color: #e74c3c; margin-top: 0;">⚡ Быстрая проверка</h3>
<ul>
    <li><strong>Браузерная консоль</strong> - F12</li>
    <li><strong>Проверка фрагментов</strong> - Direct Input</li>
    <li><strong>Расширения браузера</strong> - HTML Validator</li>
        <li><strong>Редакторы кода</strong> - Встроенная проверка</li>
            </ul>
        </div>
    </div>
</div>

<div class="note">
    <strong>🌟 Итоговый совет:</strong> Начните с W3C Validator — это официальный инструмент с самыми точными проверками.
    Используйте его для каждой страницы, которую создаёте, и Ваш код всегда будет соответствовать стандартам!
</div>
</body>
</html>
```

Часть 6: Что Вы теперь умеете

- Пользоваться W3C Validator — официальным инструментом проверки
- Выбирать подходящий способ проверки (URI, файл, прямой ввод)
- Понимать результаты проверки — ошибки и предупреждения
- Использовать альтернативные валидаторы для разных задач
- Применять браузерные инструменты для быстрой проверки
- Находить и исправлять ошибки в реальном коде

Часть 7: Итоги и рекомендации

🎯 Ключевые преимущества онлайн-валидаторов:

1. **Бесплатность** — не нужно ничего покупать
2. **Доступность** — работают из любого браузера
3. **Простота** — интуитивно понятный интерфейс
4. **Точность** — проверяют по официальным стандартам
5. **Обучение** — помогают понять ошибки и исправить их

💡 Практические советы по использованию:

Начните с W3C Validator — это золотой стандарт

Проверяйте код постепенно — не ждите окончания проекта
Читайте объяснения ошибок — это лучший способ обучения
Исправляйте ошибки по порядку — иногда одна ошибка вызывает несколько сообщений
Сохраняйте отчеты — чтобы отслеживать прогресс

 **Простые шаги для начала:**

Закладка — добавьте validator.w3.org в закладки
Привычка — проверяйте каждую новую страницу
Анализ — изучайте не только ошибки, но и предупреждения
Практика — используйте валидатор как учебное пособие

Помните: Онлайн-валидаторы — это Ваши бесплатные персональные репетиторы по HTML. Используйте их регулярно, и Вы быстро станете экспертом в создании качественного, стандартного кода! 

12.3. Как читать сообщения об ошибках и не пугаться их

Введение: Ваш друг-переводчик с "технического" на "человеческий"

Представьте, что Вы впервые приехали в другую страну и не знаете язык. Сначала всё кажется непонятным и пугающим. Но потом Вы находите переводчика, который объясняет, что на самом деле означают эти странные слова. Сообщения об ошибках — это такой же "иностранный язык", а сегодня мы станем Вашим переводчиком!

Часть 1: Почему сообщения об ошибках кажутся страшными

Давайте посмотрим на типичное сообщение об ошибке:

text

Error: End tag for li seen, but there were open elements.
Line 38, Column 25:

Что видит новичок:

Непонятные английские слова
Странные технические термины
Номера строк и столбцов
✖ Паника: "Я ничего не понимаю!"

Что видит опытный разработчик:

- ✓ Конкретное указание на проблему
- ✓ Точное место ошибки
- ✓ Чёткое объяснение
- ✓ План действий по исправлению

Часть 2: Анатомия сообщения об ошибке

Разберём типичное сообщение на части:

html

```
<!-- Пример кода с ошибкой -->
<ul>
    <li>Пункт 1
    <li>Пункт 2</li>
</ul>
```

Сообщение валидатора:

text

- ✖ Error: Unclosed element "li"
- Line 2, Column 10: Пункт 1
- Suggestion: Add closing tag

Разбираем по частям:

- ✖ Error — тип проблемы (ошибка, а не предупреждение)
- Unclosed element "li" — суть проблемы (не закрыт элемент li)
- Line 2, Column 10 — точное местоположение
- Add closing tag — конкретное решение

Часть 3: Полная страница с расшифровкой сообщений об ошибках

html

```
<!DOCTYPE html>
<html lang="ru">
<head>
  <meta charset="utf-8">
  <title>Как читать сообщения об ошибках: Расшифровываем язык
валидаторов</title>
<style>
  body {
    font-family: 'Arial', sans-serif;
    max-width: 1200px;
    margin: 0 auto;
    padding: 20px;
    line-height: 1.6;
    background-color: #f8f9fa;
  }
  .example-section {
    background: white;
    padding: 25px;
    margin: 25px 0;
    border-radius: 10px;
    box-shadow: 0 2px 5px rgba(0,0,0,0.1);
  }
  h1, h2 {
    color: #2c3e50;
  }
  h1 {
    border-bottom: 3px solid #3498db;
    padding-bottom: 15px;
  }
  .code-example {
```

```
background: #2d2d2d;
color: #f8f8f2;
padding: 15px;
border-radius: 5px;
font-family: 'Courier New', monospace;
margin: 15px 0;
overflow-x: auto;
position: relative;
}
.error-message {
background: #ffebee;
border-left: 4px solid #e74c3c;
padding: 15px;
margin: 15px 0;
border-radius: 5px;
font-family: 'Courier New', monospace;
}
.warning-message {
background: #fff3e0;
border-left: 4px solid #ff9800;
padding: 15px;
margin: 15px 0;
border-radius: 5px;
font-family: 'Courier New', monospace;
}
.success-message {
background: #e8f5e8;
border-left: 4px solid #4CAF50;
padding: 15px;
margin: 15px 0;
border-radius: 5px;
}
.note {
background: #e7f3ff;
padding: 15px;
border-left: 4px solid #2196F3;
margin: 15px 0;
border-radius: 5px;
}
.decoder {
background: #f3e5f5;
border: 2px dashed #9c27b0;
border-radius: 8px;
padding: 20px;
margin: 20px 0;
```

```
}

.message-breakdown {
    display: grid;
    grid-template-columns: 100px 1fr;
    gap: 10px;
    align-items: start;
    margin: 15px 0;
}

.message-part {
    background: #e3f2fd;
    padding: 8px 12px;
    border-radius: 5px;
    font-weight: bold;
}

.message-explanation {
    padding: 8px 12px;
}

.error-type {
    display: inline-block;
    padding: 4px 8px;
    border-radius: 4px;
    font-weight: bold;
    margin-right: 10px;
}

.error-critical {
    background: #ffccdd2;
    color: #c62828;
}

.error-warning {
    background: #ffe0b2;
    color: #ef6c00;
}

.translation-card {
    background: white;
    border: 1px solid #e0e0e0;
    border-radius: 8px;
    padding: 0;
    margin: 20px 0;
    overflow: hidden;
}

.original-message {
    background: #2c3e50;
    color: white;
    padding: 15px;
    font-family: 'Courier New', monospace;
```

```
        }
    .human-translation {
        background: #e8f5e8;
        padding: 15px;
        border-left: 4px solid #4CAF50;
    }
    .practice-area {
        background: #fff3e0;
        padding: 20px;
        border-radius: 8px;
        margin: 20px 0;
    }
    .interactive-demo {
        background: #f5f5f5;
        border: 2px solid #ddd;
        border-radius: 8px;
        padding: 20px;
        margin: 20px 0;
    }
    .error-dictionary {
        display: grid;
        grid-template-columns: repeat(auto-fit, minmax(300px, 1fr));
        gap: 15px;
        margin: 20px 0;
    }
    .dictionary-item {
        background: white;
        border: 1px solid #e0e0e0;
        border-radius: 8px;
        padding: 15px;
    }
</style>
</head>
<body>
    <h1> Как читать сообщения об ошибках и не пугаться их</h1>

    <div class="note">
        <strong> Основная идея:</strong> Сообщения об ошибках — это не критика, а помощь! Они как дорожные знаки: показывают, где нужно быть внимательным и что исправить, чтобы безопасно достичь цели.
    </div>

    <!-- Психология ошибок -->
    <div class="example-section">
        <h2>1. Почему мы пугаемся ошибок? Психологический аспект</h2>
```

```
<div style="display: grid; grid-template-columns: repeat(auto-fit, minmax(300px, 1fr)); gap: 20px; margin: 20px 0;">
    <div style="border: 2px solid #e74c3c; padding: 20px; border-radius: 8px;">
        <h3 style="color: #e74c3c; margin-top: 0;">😢 Страх новичка</h3>
        <ul>
            <li>"Я сделал что-то неправильно"</li>
            <li>"Это слишком сложно для меня"</li>
            <li>"Я никогда не пойму эти термины"</li>
            <li>"Лучше бросить, чем разбираться"</li>
        </ul>
    </div>

    <div style="border: 2px solid #3498db; padding: 20px; border-radius: 8px;">
        <h3 style="color: #3498db; margin-top: 0;">😊 Реальность</h3>
        <ul>
            <li>Ошибки — нормальная часть обучения</li>
            <li>Даже эксперты постоянно видят ошибки</li>
            <li>Каждая ошибка — возможность научиться</li>
            <li>Сообщения созданы, чтобы помочь вам</li>
        </ul>
    </div>
</div>

<div class="note">
    <strong>🎯 Помните:</strong> Каждая найденная ошибка — это шаг вперёд в Вашем обучении. Профессионалы не те, кто не делает ошибок, а те, кто умеет их быстро находить и исправлять!
</div>
</div>

<!-- Анатомия сообщения об ошибке --&gt;
&lt;div class="example-section"&gt;
    &lt;h2&gt;2. Анатомия сообщения об ошибке: Разбираем на части&lt;/h2&gt;

    &lt;div class="interactive-demo"&gt;
        &lt;h3&gt;🔍 Рассмотрим реальное сообщение об ошибке:&lt;/h3&gt;

        &lt;div class="error-message"&gt;
            &lt;span class="error-type error-critical"&gt;ERROR&lt;/span&gt;
            End tag for "p" missing, but its declaration does not permit this.&lt;br&gt;
            &lt;strong&gt;Line 5, Column 3:&lt;/strong&gt; &lt;div&gt;&lt;br&gt;</pre>
```

```
<em>Info: You probably forgot to close the "p" element.</em>
</div>

<div class="message-breakdown">
  <div class="message-part">ERROR</div>
  <div class="message-explanation">
    <strong>Тип проблемы:</strong> Это ошибка (требует исправления), а не предупреждение
  </div>

  <div class="message-part">Сообщение</div>
  <div class="message-explanation">
    <strong>Суть проблемы:</strong> "Не хватает закрывающего тега для 'p', но его объявление это не разрешает"
  </div>

  <div class="message-part">Line 5, Column 3</div>
  <div class="message-explanation">
    <strong>Местоположение:</strong> Ошибка в 5-й строке, в 3-м символе (считая слева)
  </div>

  <div class="message-part">Info</div>
  <div class="message-explanation">
    <strong>Подсказка:</strong> "Вы, вероятно, забыли закрыть элемент 'p'"
  </div>
  </div>
</div>

<div class="decoder">
  <h3>✖ Что было в коде:</h3>
  <div class="code-example">
<!DOCTYPE html>
<html>
<head>
  <title>Пример</title>
</head>
<body>
  <p>Этот абзац не закрыт
  <div>
    Другой контент
  </div>
</body>
</html>
```

```
</div>

<h3>🔧 Как исправить:</h3>
<div class="code-example">
<!DOCTYPE html>
<html>
<head>
    <title>Пример</title>
</head>
<body>
    <p>Этот абзац не закрыт</p>
    <div>
        Другой контент
    </div>
</body>
</html>
    </div>
    </div>
</div>


<div class="example-section">
    <h2>3. Словарь распространённых сообщений об ошибках</h2>

    <div class="error-dictionary">
        <div class="dictionary-item">
            <h4>🚫 "End tag for ... missing"</h4>
            <p><strong>Перевод:</strong> "Не хватает закрывающего тега для ..."</p>
            <p><strong>Пример:</strong> <code><p>Текст</code></p><br><strong>Исправление:</strong> Добавить<br><code></p></code></p>
        </div>

        <div class="dictionary-item">
            <h4>🚫 "Element ... not allowed as child of element ..."</h4>
            <p><strong>Перевод:</strong> "Элемент ... нельзя размещать внутри элемента ..."</p>
            <p><strong>Пример:</strong> <code><p><div>нельзя</div></p></code></p>
            <p><strong>Исправление:</strong> Изменить структуру вложенности</p>
        </div>

        <div class="dictionary-item">
```

```
<h4>⚠ "Consider adding alt attribute"</h4>
<p><strong>Перевод:</strong> "Рекомендуется добавить
атрибут alt"</p>
<p><strong>Пример:</strong> <code></code></p>
<p><strong>Исправление:</strong> <code></code></p>
</div>

<div class="dictionary-item">
<h4>🚫 "Duplicate attribute ..."</h4>
<p><strong>Перевод:</strong> "Повторяющийся атрибут ..."</p>
<p><strong>Пример:</strong> <code><div class="red"
class="big"></code></p>
<p><strong>Исправление:</strong> <code><div class="red
big"></code></p>
</div>

<div class="dictionary-item">
<h4>🚫 "Attribute ... not allowed on element ..."</h4>
<p><strong>Перевод:</strong> "Атрибут ... нельзя использовать
с элементом ..."</p>
<p><strong>Пример:</strong> <code><span
src="image.jpg"></code></p>
<p><strong>Исправление:</strong> Использовать правильный
элемент</p>
</div>

<div class="dictionary-item">
<h4>🚫 "Bad value ... for attribute ..."</h4>
<p><strong>Перевод:</strong> "Неправильное значение ... для
атрибута ..."</p>
<p><strong>Пример:</strong> <code><input type="number"
max="abc"></code></p>
<p><strong>Исправление:</strong> Использовать правильное
значение</p>
</div>
</div>
</div>

<!-- Страгегия работы с ошибками -->
<div class="example-section">
<h2>4. Пошаговая стратегия работы с ошибками</h2>
```

```
<div style="display: grid; grid-template-columns: repeat(auto-fit, minmax(250px, 1fr)); gap: 15px; margin: 20px 0;">
    <div style="border: 2px solid #3498db; padding: 20px; border-radius: 8px; text-align: center;">
        <div style="font-size: 2em; margin-bottom: 10px;">[1]</div>
        <h3 style="color: #3498db; margin-top: 0;">Не паникуйте</h3>
        <p>Ошибки — это нормально. Сделайте глубокий вдох.</p>
    </div>

    <div style="border: 2px solid #9b59b6; padding: 20px; border-radius: 8px; text-align: center;">
        <div style="font-size: 2em; margin-bottom: 10px;">[2]</div>
        <h3 style="color: #9b59b6; margin-top: 0;">Прочитайте внимательно</h3>
        <p>Не просто пробегите глазами, а прочтайте каждое слово.</p>
    </div>

    <div style="border: 2px solid #2ecc71; padding: 20px; border-radius: 8px; text-align: center;">
        <div style="font-size: 2em; margin-bottom: 10px;">[3]</div>
        <h3 style="color: #2ecc71; margin-top: 0;">Найдите место ошибки</h3>
        <p>Используйте указание строки и столбца.</p>
    </div>

    <div style="border: 2px solid #e67e22; padding: 20px; border-radius: 8px; text-align: center;">
        <div style="font-size: 2em; margin-bottom: 10px;">[4]</div>
        <h3 style="color: #e67e22; margin-top: 0;">Поймите суть</h3>
        <p>Что именно говорит сообщение? Используйте наш "переводчик".</p>
    </div>

    <div style="border: 2px solid #e74c3c; padding: 20px; border-radius: 8px; text-align: center;">
        <div style="font-size: 2em; margin-bottom: 10px;">[5]</div>
        <h3 style="color: #e74c3c; margin-top: 0;">Исправьте одну ошибку</h3>
        <p>Не пытайтесь исправить всё сразу.</p>
    </div>

    <div style="border: 2px solid #f1c40f; padding: 20px; border-radius: 8px; text-align: center;">
        <div style="font-size: 2em; margin-bottom: 10px;">[6]</div>
```

```
<h3 style="color: #f1c40f; margin-top: 0;">Проверьте снова</h3>
<p>Убедитесь, что ошибка исчезла, и переходите к
следующей.</p>
</div>
</div>
</div>
```

```
<!-- Практика: Переводчик ошибок -->
<div class="example-section">
    <h2>5. Практика: Переводчик с "технического" на
    "человеческий"</h2>
```

```
<div class="translation-card">
    <div class="original-message">
        ✗ Error: Attribute "border" not allowed on element "table" at this
        point.<br>
        ↪ Line 10, Column 15: <table border="1">
    </div>
    <div class="human-translation">
        <strong>❗️💻 Человеческий перевод:</strong><br>
        "Вы использовали атрибут 'border' у элемента 'table', но в
        современном HTML это не разрешается. Вместо этого используйте CSS
        для задания границ таблицы."
    </div>
</div>
```

```
<div class="translation-card">
    <div class="original-message">
        ✗ Error: Duplicate ID "header".<br>
        ↪ Line 5, Column 10: <div id="header"><br>
        ↪ Line 15, Column 10: <div id="header">
    </div>
    <div class="human-translation">
        <strong>❗️💻 Человеческий перевод:</strong><br>
        "Вы использовали одинаковый ID 'header' для двух разных
        элементов. ID должны быть уникальными на странице. Измените один из
        них на другое значение."
    </div>
</div>
```

```
<div class="translation-card">
    <div class="original-message">
        △ Warning: Section lacks heading. Consider using h2-h6
        elements.<br>
        ↪ Line 8, Column 3: <section>
```

```
</div>
<div class="human-translation">
    <strong>❗️💻 Человеческий перевод:</strong><br>
    "В секции <section> нет заголовка. Рекомендуется добавить
заголовок (h2, h3, h4, h5 или h6) в начало секции для лучшей структуры
и доступности."
    </div>
</div>
```

<!-- Работа с несколькими ошибками -->

```
<div class="example-section">
    <h2>6. Как работать с несколькими ошибками одновременно</h2>
```

```
<div class="note">
    <strong>🎯 Важная стратегия:</strong> Когда валидатор
показывает 10+ ошибок, не пытайтесь исправить всё сразу.
    Часто одна основная ошибка вызывает несколько вторичных.
</div>
```

```
<div class="interactive-demo">
    <h3>📋 Пример: Цепочка ошибок</h3>
```

```
<div class="code-example">
<!DOCTYPE html>
<html>
<head>
    <title>Мой сайт</title>
</head>
<body>
    <div>
        <p>Первый абзац
        <p>Второй абзац
        <ul>
            <li>Пункт 1
            <li>Пункт 2
        </ul>
    <div>
</body>
</html>
</div>
```

```
<div class="error-message">
    <strong>Сообщения валидатора:</strong><br>
    1. ❌ End tag for "p" missing<br>
```

```
2. ✗ End tag for "p" missing <br>
3. ✗ End tag for "li" missing<br>
4. ✗ End tag for "li" missing<br>
5. ✗ End tag for "div" missing<br>
6. ✗ Unexpected end tag : div
</div>
```

```
<div class="success-message">
```

 Решение: Начните с первой ошибки. После исправления незакрытого <p> многие последующие ошибки могут исчезнуть автоматически!

```
    </div>
</div>
```

```
<div class="practice-area">
```

```
    <h3>🎯 Практический совет:</h3>
```

 Исправляйте ошибки в порядке их появления в коде, а не в порядке списка валидатора.</p>

<p>Часто ошибка в начале кода "ломает" понимание всего последующего кода браузером.</p>

```
    </div>
</div>
```

```
<!-- Практическое задание -->
```

```
<div class="example-section">
```

```
    <h2>🎯 Практическое задание: Стань супер-детективом ошибок!
```

```
</h2>
```

```
<div class="practice-task">
```

```
    <h3>Расшифруй эти сообщения об ошибках</h3>
```

```
<div class="interactive-demo">
```

```
    <h4>🔍 Сообщение 1:</h4>
```

```
    <div class="error-message">
```

 Error: Bad value "button" for attribute "type" on element "input".

 Line 7, Column 25: <input type="button">

```
</div>
```

```
<div class="practice-area">
```

```
    <strong>Вопросы для размышления:</strong>
```

```
    <ul>
```

 Что означает "Bad value"?

 Почему "button" — неправильное значение для type?

```
</li>
```

```
<li>Какие правильные значения Вы знаете для type у  
input?</li>  
    <li>Как бы Вы исправили эту ошибку?</li>  
    </ul>  
    </div>  
</div>
```

```
<div class="interactive-demo">  
    <h4>🔍 Сообщение 2:</h4>  
    <div class="warning-message">  
        △ Warning: The "name" attribute is obsolete. Consider putting an  
        "id" on the nearest container instead.<br>  
        🔑 Line 3, Column 15: <a name="top"></a>  
</div>
```

```
<div class="practice-area">  
    <strong>Вопросы для размышления:</strong>  
    <ul>  
        <li>Что означает "obsolete"?</li>  
        <li>Почему атрибут "name" устарел для ссылок?</li>  
        <li>Что рекомендуется использовать вместо этого?</li>  
        <li>Как бы Вы переписали этот код?</li>  
    </ul>  
    </div>  
</div>
```

```
<details style="margin-top: 20px;">  
    <summary><strong>💡 Подсказки и ответы:</strong></summary>  
    <div class="success-message">  
        <h4>🔍 Ответ к Сообщению 1:</h4>  
        <p>"button" — неправильное значение для type у input.
```

Правильные значения: "text", "password", "email", "submit" и др. Для кнопки используйте `<button></button>` или `<input type="submit"></input>`.

```
<h4>🔍 Ответ к Сообщению 2:</h4>  
<p>Атрибут "name" у ссылок устарел. Вместо этого  
используйте "id" у ближайшего контейнера: <code><div id="top"></code> и  
ссылайтесь через <code>href="#top"</code>. </p>  
    </div>  
    </details>  
</div>  
</div>
```

```
<!-- Шпаргалка -->
```

```
<div class="example-section">
  <h2>📋 Шпаргалка: Не паникуй! Руководство по выживанию</h2>

    <div style="display: grid; grid-template-columns: repeat(auto-fit, minmax(300px, 1fr)); gap: 15px; margin: 20px 0;">
      <div style="border: 2px solid #3498db; padding: 15px; border-radius: 8px;">
        <h3 style="color: #3498db; margin-top: 0;">❗ Психологическая подготовка</h3>
        <ul>
          <li>Ошибки — это помощь, а не критика</li>
          <li>Каждая ошибка — урок</li>
          <li>Даже гении видят ошибки</li>
          <li>Сделайте перерыв, если чувствуете разочарование</li>
        </ul>
      </div>

      <div style="border: 2px solid #27ae60; padding: 15px; border-radius: 8px;">
        <h3 style="color: #27ae60; margin-top: 0;">🔍 Техника чтения ошибок</h3>
        <ul>
          <li>Читайте медленно и внимательно</li>
          <li>Разбирайте сообщение на части</li>
          <li>Используйте указания строк и столбцов</li>
          <li>Ищите ключевые слова</li>
        </ul>
      </div>

      <div style="border: 2px solid #e74c3c; padding: 15px; border-radius: 8px;">
        <h3 style="color: #e74c3c; margin-top: 0;">⚡ Стратегия исправления</h3>
        <ul>
          <li>Исправляйте по одной ошибке за раз</li>
          <li>Начинайте с первых ошибок в коде</li>
          <li>Проверяйте после каждого исправления</li>
          <li>Не бойтесь экспериментировать</li>
        </ul>
      </div>
    </div>

    <div class="note">
      <strong>⭐ Запомните:</strong> Сообщения об ошибках — это Ваш лучший друг в обучении.
    </div>
  </div>
```

Они как личный репетитор, который всегда рядом и бесплатно помогает Вам становиться лучше!

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

Часть 7: Что Вы теперь умеете

- Понимать психологию ошибок и не бояться их
- Разбирать сообщения об ошибках на составные части
- Использовать "переводчик" с технического языка на человеческий
- Применять пошаговую стратегию работы с ошибками
- Работать с несколькими ошибками одновременно
- Распознавать распространённые типы ошибок и быстро их исправлять

Часть 8: Итоги и рекомендации

Ключевые изменения в восприятии ошибок:

От страха к любопытству — ошибки становятся загадками, которые интересно разгадывать

От растерянности к уверенности — Вы знаете, как подходить к любой ошибке

От избегания к поиску — Вы начинаете proactively искать ошибки для обучения

От зависимости к самостоятельности — Вы можете сами разобраться с большинством проблем

Практические привычки для развития:

Читайте ошибки вслух — это помогает лучше понять суть

Ведите дневник ошибок — записывайте, какие ошибки встречались и как их исправили

Празднуйте исправления — каждая исправленная ошибка это маленькая победа

Помогайте другим — объясняя ошибки другим, Вы лучше их понимаете сами

Продвинутые техники:

Предсказывайте ошибки — со временем Вы начнёте предвидеть, где могут возникнуть проблемы

Анализируйте закономерности — какие ошибки Вы делаете чаще всего?

Создайте чек-лист — список самых частых ошибок для быстрой проверки

Используйте ошибки как учебный материал — каждая ошибка учит чему-то новому о HTML

Помните: Ошибки — это не стены, которые преграждают Вам путь, а ступеньки, которые помогают подниматься выше в Вашем мастерстве веб-разработчика! 

Теперь, когда Вы умеете читать сообщения об ошибках, Вы обладаете суперсилой — способностью учиться и расти с каждой написанной строкой кода! 

12.4. Практика: проверяем свою страницу и исправляем найденные ошибки

Введение: Ваш первый техосмотр сайта

Представьте, что Вы купили подержанный автомобиль и пригнали его в сервис на полную диагностику. Мастер проверяет каждую систему, находит проблемы и составляет план ремонта. Сегодня мы проведём такую же полную диагностику Вашего HTML-кода — найдём все проблемы и планомерно их исправим!

Часть 1: Подготовка к проверке — создаём тестовую страницу

Давайте создадим страницу с типичными ошибками, которые часто делают новички:

```
html
<!DOCTYPE html>
<html>
<head>
    <title>Мой сайт о садоводстве</title>
</head>
<body>
    <h1>Добро пожаловать в мой сад!

    <p>Здесь я выращиваю разные растения

    <h2>Мои любимые цветы</h2>
    <ul>
        <li>Розы
        <li>Тюльпаны
        <li>Хризантемы
    </ul>

    <table border=1>
        <tr>
            <td>Овощ
            <td>Количество
        <tr>
            <td>Помидоры
            <td>10 кустов
    </table>
```

```
<center>Приходите в гости полюбоваться!</center>
```

```
    <font size="5">До встречи в саду!</font>
</body>
</html>
```

Часть 2: Пошаговый процесс проверки и исправления

Шаг 1: Проверяем страницу в валидаторе

Переходим на validator.w3.org → "Validate by Direct Input" → вставляем наш код → "Check"

Результат проверки:

```
text
✖ 7 ошибок, △ 2 предупреждения
```

Шаг 2: Анализируем отчёт и создаём план исправлений

```
html
<!-- Создаём таблицу для отслеживания прогресса --&gt;
<table class="error-tracker">
| № | Ошибка | Статус | Решение |
| --- | --- | --- | --- |
| 1 | No charset declared | ✖ | Добавить meta charset |
| 2 | End tag for h1 missing | ✖ | Закрыть тег h1 |
| 3 | End tag for p missing | ✖ | Закрыть тег p |

<!-- и так далее --&gt;
</table>
```

Часть 3: Полная страница с интерактивным процессом исправления

```
html
<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="utf-8">
    <title>Практика: Проверяем и исправляем ошибки в HTML</title>
    <style>
        body {
            font-family: 'Arial', sans-serif;
            max-width: 1200px;
        }
    </style>

```

```
margin: 0 auto;
padding: 20px;
line-height: 1.6;
background-color: #f8f9fa;
}
.example-section {
background: white;
padding: 25px;
margin: 25px 0;
border-radius: 10px;
box-shadow: 0 2px 5px rgba(0,0,0,0.1);
}
h1, h2 {
color: #2c3e50;
}
h1 {
border-bottom: 3px solid #3498db;
padding-bottom: 15px;
}
.code-example {
background: #2d2d2d;
color: #f8f8f2;
padding: 15px;
border-radius: 5px;
font-family: 'Courier New', monospace;
margin: 15px 0;
overflow-x: auto;
position: relative;
}
.error-code {
background: #ffebbe;
border-left: 4px solid #e74c3c;
padding: 15px;
margin: 15px 0;
border-radius: 5px;
}
.fixed-code {
background: #e8f5e8;
border-left: 4px solid #27ae60;
padding: 15px;
margin: 15px 0;
border-radius: 5px;
}
.note {
background: #e7f3ff;
```

```
padding: 15px;
border-left: 4px solid #2196F3;
margin: 15px 0;
border-radius: 5px;
}
.warning {
background: #fff3e0;
padding: 15px;
border-left: 4px solid #ff9800;
margin: 15px 0;
border-radius: 5px;
}
.success {
background: #e8f5e8;
padding: 15px;
border-left: 4px solid #4CAF50;
margin: 15px 0;
border-radius: 5px;
}
.error-marker {
background: #ffcdd2;
color: #c62828;
padding: 2px 4px;
border-radius: 3px;
font-weight: bold;
}
.fix-marker {
background: #c8e6c9;
color: #2e7d32;
padding: 2px 4px;
border-radius: 3px;
font-weight: bold;
}
.progress-tracker {
background: #f5f5f5;
border: 2px solid #ddd;
border-radius: 8px;
padding: 20px;
margin: 20px 0;
}
.error-item {
display: flex;
align-items: center;
padding: 10px;
border-bottom: 1px solid #eee;
```

```
        }
.error-item:last-child {
    border-bottom: none;
}
.error-status {
    width: 30px;
    height: 30px;
    border-radius: 50%;
    display: flex;
    align-items: center;
    justify-content: center;
    margin-right: 15px;
    font-weight: bold;
}
.status-pending {
    background: #ffccdd2;
    color: #c62828;
}
.status-fixed {
    background: #c8e6c9;
    color: #2e7d32;
}
.error-details {
    flex-grow: 1;
}
.interactive-demo {
    background: #fff3e0;
    padding: 20px;
    border-radius: 8px;
    margin: 20px 0;
}
.code-comparison {
    display: grid;
    grid-template-columns: 1fr 1fr;
    gap: 20px;
    margin: 20px 0;
}
@media (max-width: 768px) {
    .code-comparison {
        grid-template-columns: 1fr;
    }
}
.before-box, .after-box {
    padding: 15px;
    border-radius: 5px;
```

```
        }
    .before-box {
        background: #ffebee;
        border: 1px solid #e74c3c;
    }
    .after-box {
        background: #e8f5e8;
        border: 1px solid #27ae60;
    }
    .checkpoint {
        background: #e3f2fd;
        border-left: 4px solid #2196F3;
        padding: 15px;
        margin: 15px 0;
        border-radius: 5px;
    }

```

</style>

</head>

<body>

<h1>🔧 Практика: Проверяем свою страницу и исправляем найденные ошибки</h1>

<div class="note">

💡 Основная идея: Сегодня мы проведём полный "техосмотр" HTML-страницы. Мы найдём все ошибки, планомерно их исправим и превратим проблемный код в идеально валидный!

</div>

<!-- Исходный код для проверки -->

<div class="example-section">

<h2>1. Наша тестовая страница с ошибками</h2>

<div class="warning">

⚠ Внимание! Этот код содержит несколько типичных ошибок, которые часто делают новички. Наша задача — найти и исправить их все!

</div>

<div class="code-example">

```
<!DOCTYPE html>
<html>
<head>
    <title>Мой сайт о садоводстве</title>
</head>
<body>
```

```
<h1>Добро пожаловать в мой сад!
<p>Здесь я выращиваю разные растения
<h2>Мои любимые цветы</h2>
<ul>
    <li>Розы
    <li>Тюльпаны
    <li>Хризантемы
</ul>



<table border=1>
    <tr>
        <td>Овощ
        <td>Количество
    <tr>
        <td>Помидоры
        <td>10 кустов
</table>

<center>Приходите в гости полюбоваться!</center>

<font size="5">До встречи в саду!</font>
</body>
</html>
</div>

<div class="interactive-demo">
    <h3>🎯 Ваша задача:</h3>
    <ol>
        <li>Скопируйте код выше</li>
        <li>Перейдите на <a href="https://validator.w3.org"
target="_blank">validator.w3.org</a></li>
        <li>Вставьте код и запустите проверку</li>
        <li>Запишите все найденные ошибки</li>
        <li>Сверьтесь с нашим анализом ниже</li>
    </ol>
</div>
</div>

<!-- Анализ ошибок -->
<div class="example-section">
    <h2>2. Анализ ошибок: Что нашёл валидатор?</h2>
```

```
<div class="progress-tracker">
  <h3>📋 Отчёт о проверке:</h3>

  <div class="error-item">
    <div class="error-status status-pending">1</div>
    <div class="error-details">
      <strong>✖ Error:</strong> No charset declared<br>
      <em>Location:</em> Весь документ<br>
      <strong>Решение:</strong> Добавить <code><meta charset="utf-8"></code>
    </div>
  </div>

  <div class="error-item">
    <div class="error-status status-pending">2</div>
    <div class="error-details">
      <strong>✖ Error:</strong> End tag for h1 missing<br>
      <em>Location:</em> Line 6<br>
      <strong>Решение:</strong> Добавить <code></h1></code>
    </div>
  </div>

  <div class="error-item">
    <div class="error-status status-pending">3</div>
    <div class="error-details">
      <strong>✖ Error:</strong> End tag for p missing<br>
      <em>Location:</em> Line 8<br>
      <strong>Решение:</strong> Добавить <code></p></code>
    </div>
  </div>

  <div class="error-item">
    <div class="error-status status-pending">4</div>
    <div class="error-details">
      <strong>✖ Error:</strong> End tag for li missing (3 раза)<br>
      <em>Location:</em> Lines 11-13<br>
      <strong>Решение:</strong> Добавить <code></li></code> для
    каждого пункта
    </div>
  </div>

  <div class="error-item">
    <div class="error-status status-pending">5</div>
    <div class="error-details">
```

```
<strong>⚠ Warning:</strong> Consider adding alt attribute<br>
<em>Location:</em> Line 15<br>
<strong>Решение:</strong> Добавить
<code>alt="Описание"</code>
    </div>
</div>
```

```
<div class="error-item">
    <div class="error-status status-pending">6</div>
    <div class="error-details">
        <strong>✖ Error:</strong> Attribute border not allowed on
table<br>
        <em>Location:</em> Line 17<br>
        <strong>Решение:</strong> Удалить border, использовать
```

CSS

```
    </div>
</div>

<div class="error-item">
    <div class="error-status status-pending">7</div>
    <div class="error-details">
        <strong>✖ Error:</strong> End tag for td missing (4 раза)<br>
        <em>Location:</em> Lines 18-22<br>
        <strong>Решение:</strong> Добавить <code></td></code> и
<code></tr></code>
    </div>
</div>

<div class="error-item">
    <div class="error-status status-pending">8</div>
    <div class="error-details">
        <strong>✖ Error:</strong> Element center not allowed<br>
        <em>Location:</em> Line 24<br>
        <strong>Решение:</strong> Заменить на CSS <code>text-
align: center</code>
    </div>
</div>
```

```
<div class="error-item">
    <div class="error-status status-pending">9</div>
    <div class="error-details">
        <strong>✖ Error:</strong> Element font not allowed<br>
        <em>Location:</em> Line 26<br>
        <strong>Решение:</strong> Заменить на CSS
    </div>
```

```
</div>
</div>
</div>

<!-- Процесс исправления -->
<div class="example-section">
    <h2>3. Пошаговый процесс исправления ошибок</h2>

    <div class="checkpoint">
        <h3>🎯 Стратегия:</h3>
        <p>Исправляем ошибки <strong>сверху вниз</strong> — так
последующие ошибки часто исправляются автоматически!</p>
    </div>

    <!-- Ошибка 1: Charset -->
    <div class="code-comparison">
        <div class="before-box">
            <h4>❌ Было (Ошибка 1):</h4>
            <div class="code-example">
<!DOCTYPE html>
<html>
<head>
    <title>Мой сайт о садоводстве</title>
</head>
            </div>
            <p><span class="error-marker">Проблема:</span> Не указана
кодировка символов</p>
        </div>
        <div class="after-box">
            <h4>✅ Стало:</h4>
            <div class="code-example">
<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="utf-8">
    <title>Мой сайт о садоводстве</title>
</head>
            </div>
            <p><span class="fix-marker">Исправление:</span> Добавили
кодировку и язык</p>
        </div>
    </div>

    <!-- Ошибка 2-4: Незакрытые теги -->
    <div class="code-comparison">
```

```
<div class="before-box">
  <h4>☒ Было (Ошибки 2-4):</h4>
  <div class="code-example">
<h1>Добро пожаловать в мой сад!

<p>Здесь я выращиваю разные растения

<h2>Мои любимые цветы</h2>
<ul>
  <li>Розы
  <li>Тюльпаны
  <li>Хризантемы
</ul>
</div>
<p><span class="error-marker">Проблема:</span> 6 незакрытых
тегов!</p>
</div>
<div class="after-box">
  <h4>✓ Стало:</h4>
  <div class="code-example">
<h1>Добро пожаловать в мой сад!</h1>

<p>Здесь я выращиваю разные растения</p>

<h2>Мои любимые цветы</h2>
<ul>
  <li>Розы</li>
  <li>Тюльпаны</li>
  <li>Хризантемы</li>
</ul>
</div>
<p><span class="fix-marker">Исправление:</span> Закрыли все
теги</p>
</div>
</div>

<!-- Ошибка 5: Alt атрибут -->
<div class="code-comparison">
  <div class="before-box">
    <h4>☒ Было (Ошибка 5):</h4>
    <div class="code-example">

</div>
    <p><span class="error-marker">Проблема:</span> Нет alt
атрибута</p>
```

```
</div>
<div class="after-box">
    <h4>✓ Стало:</h4>
    <div class="code-example">

    </div>
    <p><span class="fix-marker">Исправление:</span> Добавили
описание изображения</p>
    </div>
</div>

<!-- Ошибка 6-7: Таблица --&gt;
&lt;div class="code-comparison"&gt;
    &lt;div class="before-box"&gt;
        &lt;h4&gt;✗ Было (Ошибки 6-7):&lt;/h4&gt;
        &lt;div class="code-example"&gt;
&lt;table border=1&gt;
    &lt;tr&gt;
        &lt;td&gt;Овощ
        &lt;td&gt;Количество
    &lt;tr&gt;
        &lt;td&gt;Помидоры
        &lt;td&gt;10 кустов
&lt;/table&gt;
        &lt;/div&gt;
        &lt;p&gt;&lt;span class="error-marker"&gt;Проблема:&lt;/span&gt; Устаревший
border и незакрытые теги&lt;/p&gt;
    &lt;/div&gt;
    &lt;div class="after-box"&gt;
        &lt;h4&gt;✓ Стало:&lt;/h4&gt;
        &lt;div class="code-example"&gt;
&lt;table class="garden-table"&gt;
    &lt;tr&gt;
        &lt;td&gt;Овощ&lt;/td&gt;
        &lt;td&gt;Количество&lt;/td&gt;
    &lt;/tr&gt;
    &lt;tr&gt;
        &lt;td&gt;Помидоры&lt;/td&gt;
        &lt;td&gt;10 кустов&lt;/td&gt;
    &lt;/tr&gt;
&lt;/table&gt;

&lt;style&gt;
.garden-table {
    border: 1px solid #ccc;</pre>
```

```
    border-collapse: collapse;
}
.garden-table td {
    border: 1px solid #ccc;
    padding: 8px;
}
</style>
</div>
<p><span class="fix-marker">Исправление:</span> Убрали
border, закрыли теги, добавили CSS</p>
</div>
</div>

<!-- Ошибка 8-9: Устаревшие теги --&gt;
&lt;div class="code-comparison"&gt;
    &lt;div class="before-box"&gt;
        &lt;h4&gt;☒ Было (Ошибки 8-9):&lt;/h4&gt;
        &lt;div class="code-example"&gt;
&lt;center&gt;Приходите в гости полюбоваться!&lt;/center&gt;

&lt;font size="5"&gt;До встречи в саду!&lt;/font&gt;
    &lt;/div&gt;
    &lt;p&gt;&lt;span class="error-marker"&gt;Проблема:&lt;/span&gt; Устаревшие
теги&lt;/p&gt;
    &lt;/div&gt;
    &lt;div class="after-box"&gt;
        &lt;h4&gt;☑ Стало:&lt;/h4&gt;
        &lt;div class="code-example"&gt;
&lt;div class="center-text"&gt;Приходите в гости полюбоваться!&lt;/div&gt;

&lt;div class="large-text"&gt;До встречи в саду!&lt;/div&gt;

&lt;style&gt;
.center-text {
    text-align: center;
}
.large-text {
    font-size: 1.5em;
}
&lt;/style&gt;
&lt;/div&gt;
&lt;p&gt;&lt;span class="fix-marker"&gt;Исправление:&lt;/span&gt; Заменили на
CSS стили&lt;/p&gt;
&lt;/div&gt;
&lt;/div&gt;</pre>
```

```
</div>

<!-- Итоговый исправленный код -->
<div class="example-section">
    <h2>4. Итоговый исправленный код</h2>

    <div class="success">
        <strong>🎉 Поздравляем!</strong> После всех исправлений мы
    получили идеально валидный HTML-код!
    </div>

    <div class="fixed-code">
        <h3>✓ Исправленная версия:</h3>
        <div class="code-example">
<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="utf-8">
    <title>Мой сайт о садоводстве</title>
    <style>
        .garden-table {
            border: 1px solid #ccc;
            border-collapse: collapse;
            margin: 20px 0;
        }
        .garden-table td {
            border: 1px solid #ccc;
            padding: 8px 12px;
        }
        .center-text {
            text-align: center;
            margin: 15px 0;
        }
        .large-text {
            font-size: 1.5em;
            color: #2c3e50;
        }
    </style>
</head>
<body>
    <h1>Добро пожаловать в мой сад!</h1>

    <p>Здесь я выращиваю разные растения</p>
    <h2>Мои любимые цветы</h2>
```

```
<ul>
  <li>Розы</li>
  <li>Тюльпаны</li>
  <li>Хризантемы</li>
</ul>



<table class="garden-table">
  <tr>
    <td>Овощ</td>
    <td>Количество</td>
  </tr>
  <tr>
    <td>Помидоры</td>
    <td>10 кустов</td>
  </tr>
</table>

<div class="center-text">Приходите в гости полюбоваться!</div>

  <div class="large-text">До встречи в саду!</div>
</body>
</html>
      </div>
    </div>

<div class="interactive-demo">
  <h3>🎯 Финальная проверка:</h3>
  <ol>
    <li>Скопируйте исправленный код выше</li>
    <li>Проверьте его в <a href="https://validator.w3.org"
target="_blank">валидаторе</a></li>
      <li>Убедитесь, что получили сообщение: <strong>"Document
checking completed. No errors or warnings to show."</strong></li>
    </ol>
  </div>
</div>

<!-- Чеклист для будущих проверок -->
<div class="example-section">
  <h2>5. Чеклист для самостоятельных проверок</h2>

  <div style="display: grid; grid-template-columns: repeat(auto-fit,
minmax(300px, 1fr)); gap: 15px; margin: 20px 0;">
```

```
<div style="border: 2px solid #3498db; padding: 15px; border-radius: 8px;">
```

```
    <h3 style="color: #3498db; margin-top: 0;"> Обязательные  
элементы</h3>
```

```
    <ul>
```

-  <code><!DOCTYPE html></code>
-  <code><html lang="ru"></code>
-  <code><meta charset="utf-8"></code>
-  <code><title>...</title></code>

```
    </ul>
```

```
    </div>
```

```
<div style="border: 2px solid #27ae60; padding: 15px; border-radius: 8px;">
```

```
    <h3 style="color: #27ae60; margin-top: 0;"> Структура  
тегов</h3>
```

```
    <ul>
```

-  Все теги закрыты
-  Правильная вложенность
-  Нет устаревших тегов
-  Уникальные ID

```
    </ul>
```

```
    </div>
```

```
<div style="border: 2px solid #e74c3c; padding: 15px; border-radius: 8px;">
```

```
    <h3 style="color: #e74c3c; margin-top: 0;"> Изображения и  
ссылки</h3>
```

```
    <ul>
```

-  <code>alt</code> у всех изображений
-  Корректные пути к файлам
-  Правильные атрибуты href
-  Доступные описания

```
    </ul>
```

```
    </div>
```

```
    </div>
```

```
<div class="note">
```

```
    <strong> Профессиональный совет:</strong> Создайте себе  
такой чеклист и используйте его каждый раз перед финальной  
проверкой в валидаторе. Это сэкономит Вам много времени!
```

```
    </div>
```

```
</div>
```

```
<!-- Практическое задание -->
```

```
<div class="example-section">
  <h2> Практическое задание: Проверь свою страницу!</h2>

  <div class="practice-task">
    <h3>Примени полученные знания на практике</h3>

    <div class="interactive-demo">
      <h4> Задание:</h4>
      <ol>
        <li><strong>Выбери свою страницу:</strong> Возьми любой HTML-файл, который ты создавал ранее</li>
        <li><strong>Проверь в валидаторе:</strong> Используй validator.w3.org</li>
        <li><strong>Составь список ошибок:</strong> Запиши все найденные проблемы</li>
        <li><strong>Исправь по порядку:</strong> Применяй стратегию "сверху вниз"</li>
        <li><strong>Добейся идеала:</strong> Убедись, что нет ни ошибок, ни предупреждений</li>
      </ol>

      <div class="progress-tracker">
        <h4> Шаблон для отслеживания прогресса:</h4>
        <textarea style="width: 100%; height: 200px; padding: 10px; border: 1px solid #ddd; border-radius: 5px; font-family: monospace;" placeholder="Создай здесь свой список ошибок:
1. [ ] Ошибка: ... | Решение: ...
2. [ ] Ошибка: ... | Решение: ...
3. [ ] Предупреждение: ... | Решение: ...
..."></textarea>
      </div>
    </div>

    <div class="note">
      <strong> Бонусное задание:</strong> После исправления всех ошибок создай короткий отчёт:
      <ul>
        <li>Сколько ошибок было изначально?</li>
        <li>Какие типы ошибок встречались чаще всего?</li>
        <li>Какая ошибка была самой сложной для исправления?</li>
      </ul>
      <li>Что ты узнал нового в процессе?</li>
    </div>
  </div>

```

```
</div>

<!-- Итоги и советы -->
<div class="example-section">
    <h2>6. Итоги и профессиональные советы</h2>

    <div style="display: grid; grid-template-columns: repeat(auto-fit, minmax(250px, 1fr)); gap: 15px; margin: 20px 0;">
        <div style="border: 2px solid #3498db; padding: 15px; border-radius: 8px; text-align: center;">
            <div style="font-size: 2em;">⟳</div>
            <h4>Регулярность</h4>
            <p>Проверяй код после каждого значительного изменения</p>
        </div>

        <div style="border: 2px solid #27ae60; padding: 15px; border-radius: 8px; text-align: center;">
            <div style="font-size: 2em;">📝</div>
            <h4>Документирование</h4>
            <p>Веди дневник частых ошибок и их решений</p>
        </div>

        <div style="border: 2px solid #e74c3c; padding: 15px; border-radius: 8px; text-align: center;">
            <div style="font-size: 2em;">🎯</div>
            <h4>Приоритеты</h4>
            <p>Сначала исправляй ошибки, потом — предупреждения</p>
        </div>

        <div style="border: 2px solid #f39c12; padding: 15px; border-radius: 8px; text-align: center;">
            <div style="font-size: 2em;">🔧</div>
            <h4>Инструменты</h4>
            <p>Используй валидацию в редакторе кода</p>
        </div>
    </div>

    <div class="success">
        <strong>🎓 Поздравляю с завершением практики!</strong>
    Теперь ты умеешь не только находить ошибки в HTML-коде, но и эффективно их исправлять. Этот навык сделает тебя настоящим профессионалом веб-разработки!
        </div>
    </div>
</body>
```

</html>

Часть 7: Что Вы теперь умеете

- Проводить полную проверку HTML-страницы в валидаторе
- Анализировать отчёты об ошибках и составлять план исправлений
- Применять стратегию исправления "сверху вниз"
- Исправлять типичные ошибки новичков
- Заменять устаревшие теги на современные аналоги
- Использовать чеклисты для систематических проверок
- Добиваться идеально валидного кода без ошибок и предупреждений

Часть 8: Итоги и профессиональные рекомендации

Ключевые результаты практики:

От хаоса к системе — Вы научились планомерному подходу к исправлению ошибок

От страха к уверенности — Вы понимаете, что любую ошибку можно исправить

От случайности к предсказуемости — Вы знаете, какие ошибки искать в первую очередь

От новичка к профессиональному — Вы приобрели навык, который используют все опытные разработчики

Профессиональные привычки для закрепления:

Проверяйте код в процессе написания — не откладывайте на потом

Создайте персональный чеклист на основе своих частых ошибок

Используйте валидацию в редакторе — настройте автоматическую проверку

Помогайте другим — объясняя ошибки, Вы лучше их запоминаете

Анализируйте закономерности — какие ошибки Вы делаете чаще всего?

Продвинутые техники для дальнейшего развития:

Интеграция со сборщиками проектов — автоматическая валидация при сборке

Настройка крючков для предварительной фиксации — проверка кода перед сохранением

Использование линтеров — инструментов для проверки стиля кода

Валидация доступности — проверка для людей с ограниченными возможностями

Аудит эффективности — проверка производительности страницы

Помните: Идеально валидный код — это не самоцель, а показатель профессионального подхода к веб-разработке. Каждая исправленная ошибка делает интернет чуточку лучше! 

Теперь Вы вооружены знаниями и инструментами, чтобы создавать безупречный HTML-код, который будет работать корректно во всех браузерах и устройствах! 

Глава 13: Как показать свой сайт миру?

13.1. Что такое хостинг и домен (простыми словами)

Введение: Ваш личный участок в интернете

Представьте, что Вы хотите построить дом. Вам нужен участок земли (хостинг) и адрес, по которому этот дом можно найти (домен). Сегодня мы разберём, как эти понятия работают в мире интернета!

Часть 1: Простая аналогия — строительство дома

Давайте проведём параллель с реальным миром:

html

```
<!-- Аналогия: Строительство дома в реальном мире -->
<div class="analogy">
  <div class="real-world">
     Реальный мир:
    - Участок земли = Хостинг
    - Адрес дома = Домен
    - Дом = Ваш сайт
    - Строительные материалы = HTML/CSS код
  </div>

  <div class="internet-world">
     Интернет:
    - Сервер = Хостинг
    - example.ru = Домен
    - Веб-сайт = Ваш сайт
    - Файлы сайта = HTML/CSS код
  </div>
</div>
```

Часть 2: Что такое домен? Ваш адрес в интернете

2.1. Домен — это Ваш уникальный адрес

html

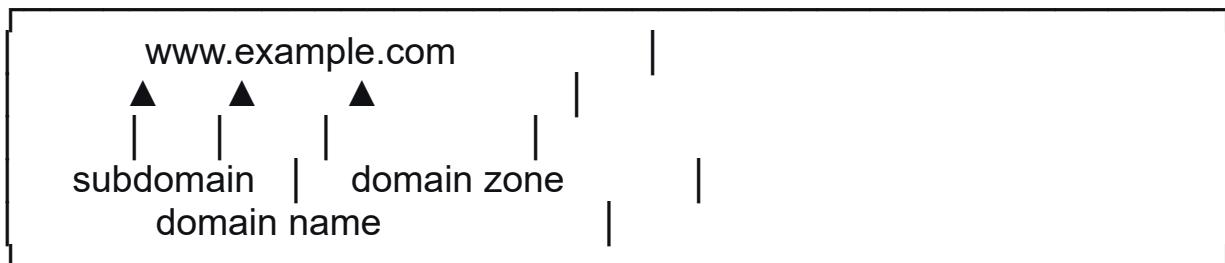
```
<div class="domain-explanation">
  <h3> Домен — это как номер телефона или домашний адрес:</h3>

  <div class="example">
    <strong>Примеры доменов:</strong>
```

```
google.com  
yandex.ru  
moy-sait.ru  
pensioner-html.ru  
</div>  
</div>
```

2.2. Из чего состоит домен:

text



Простыми словами:

example — название Вашего сайта (как название магазина)

.com — зона домена (как район города)

www — поддомен (как отдел в магазине)

Часть 3: Что такое хостинг? Ваша земля под строительство

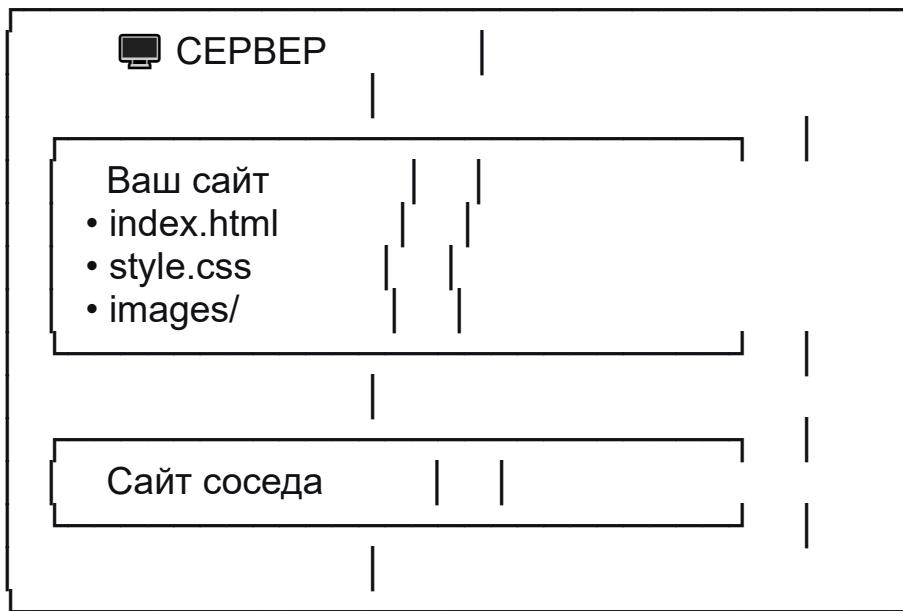
3.1. Хостинг — это место, где живёт Ваш сайт

html

```
<div class="hosting-explanation">  
    <h3>💻 Хостинг — это как аренда помещения под магазин:</h3>  
  
    <div class="analogy-list">  
        <div class="analogy-item">  
            <strong>Реальный мир:</strong> Аренда помещения в торговом  
центре  
        </div>  
        <div class="analogy-item">  
            <strong>Интернет:</strong> Аренда места на сервере у хостинг-  
компании  
        </div>  
    </div>  
</div>
```

****3.2. Что такое сервер:**

text



Часть 4: Полная страница с объяснениями и примерами

html

```
<!DOCTYPE html>
<html lang="ru">
<head>
  <meta charset="utf-8">
  <title>Хостинг и домен: Ваш участок в интернете</title>
  <style>
    body {
      font-family: 'Arial', sans-serif;
      max-width: 1200px;
      margin: 0 auto;
      padding: 20px;
      line-height: 1.6;
      background-color: #f8f9fa;
    }
    .example-section {
      background: white;
      padding: 25px;
      margin: 25px 0;
      border-radius: 10px;
      box-shadow: 0 2px 5px rgba(0,0,0,0.1);
    }
    h1, h2 {
      color: #2c3e50;
    }
    h1 {
```

```
border-bottom: 3px solid #3498db;
padding-bottom: 15px;
}
.code-example {
background: #2d2d2d;
color: #f8f8f2;
padding: 15px;
border-radius: 5px;
font-family: 'Courier New', monospace;
margin: 15px 0;
overflow-x: auto;
}
.note {
background: #e7f3ff;
padding: 15px;
border-left: 4px solid #2196F3;
margin: 15px 0;
border-radius: 5px;
}
.warning {
background: #fff3e0;
padding: 15px;
border-left: 4px solid #ff9800;
margin: 15px 0;
border-radius: 5px;
}
.success {
background: #e8f5e8;
padding: 15px;
border-left: 4px solid #4CAF50;
margin: 15px 0;
border-radius: 5px;
}
.analogy-box {
background: #f3e5f5;
border: 2px dashed #9c27b0;
border-radius: 8px;
padding: 20px;
margin: 20px 0;
}
.comparison-table {
width: 100%;
border-collapse: collapse;
margin: 20px 0;
}
```

```
.comparison-table th, .comparison-table td {
    border: 1px solid #ddd;
    padding: 15px;
    text-align: left;
}
.comparison-table th {
    background: #34495e;
    color: white;
}
.comparison-table tr:nth-child(even) {
    background: #f9f9f9;
}
.visual-explanation {
    display: grid;
    grid-template-columns: repeat(auto-fit, minmax(300px, 1fr));
    gap: 20px;
    margin: 20px 0;
}
.visual-card {
    background: white;
    border: 2px solid #3498db;
    border-radius: 10px;
    padding: 20px;
    text-align: center;
}
.visual-icon {
    font-size: 3em;
    margin-bottom: 15px;
}
.domain-structure {
    background: #e3f2fd;
    padding: 20px;
    border-radius: 8px;
    margin: 20px 0;
    text-align: center;
    font-family: 'Courier New', monospace;
    font-size: 1.2em;
}
.hosting-types {
    display: grid;
    grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));
    gap: 15px;
    margin: 20px 0;
}
.hosting-card {
```

```
background: white;
border: 1px solid #e0e0e0;
border-radius: 8px;
padding: 20px;
}
.hosting-card h3 {
color: #2c3e50;
margin-top: 0;
display: flex;
align-items: center;
gap: 10px;
}
.step-by-step {
background: #fff3e0;
padding: 20px;
border-radius: 8px;
margin: 20px 0;
}
.step {
margin: 15px 0;
padding-left: 20px;
position: relative;
}
.step::before {
content: "✓";
position: absolute;
left: 0;
color: #27ae60;
font-weight: bold;
}
</style>
</head>
<body>
<h1> Хостинг и домен: Ваш личный участок в интернете</h1>

<div class="note">
<strong> Основная идея:</strong> Хостинг и домен — это два кита, на которых стоит любой сайт в интернете. Без них Ваш прекрасный HTML-код будет просто набором файлов на компьютере, который никто не увидит.
</div>

<!-- Основная аналогия -->
<div class="example-section">
<h2>1. Основная аналогия: Строительство дома</h2>
```

```
<div class="analogy-box">
  <h3>🏠 Представьте, что Вы строите дом:</h3>

  <table class="comparison-table">
    <tr>
      <th>Реальный мир</th>
      <th>Мир интернета</th>
      <th>Объяснение</th>
    </tr>
    <tr>
      <td>Участок земли</td>
      <td>Хостинг</td>
      <td>Место, где стоит Ваш дом (сайт)</td>
    </tr>
    <tr>
      <td>Адрес дома: ул. Ленина, 15</td>
      <td>Домен: moy-sait.ru</td>
      <td>По этому адресу люди находят Ваш дом</td>
    </tr>
    <tr>
      <td>Сам дом с мебелью</td>
      <td>Файлы Вашего сайта</td>
      <td>То, что видят посетители внутри</td>
    </tr>
    <tr>
      <td>Строительные материалы</td>
      <td>HTML, CSS, JavaScript код</td>
      <td>Из чего сделан Ваш сайт</td>
    </tr>
  </table>
</div>

<div class="visual-explanation">
  <div class="visual-card">
    <div class="visual-icon">💻 </div>
    <h3>Хостинг</h3>
    <p>Это компьютер (сервер), который всегда включен и подключен к интернету. На нём хранятся все файлы Вашего сайта.</p>
  </div>

  <div class="visual-card">
    <div class="visual-icon">🌐 </div>
    <h3>Домен</h3>
  </div>

```

<p>Это красивый адрес, который люди вводят в браузере, чтобы найти Ваш сайт. Без домена пришлось бы запоминать цифровые IP-адреса.</p>

</div>

<div class="visual-card">
 <div class="visual-icon">🔗</div>
 <h3>Связь</h3>
 <p>Домен "указывает" на хостинг. Когда кто-то вводит Ваш домен, интернет направляет его на Ваш хостинг, где лежат файлы сайта.</p>
 </div>
</div>
</div>

<!-- Подробно о доменах -->

<div class="example-section">
 <h2>2. Что такое домен? Ваш адрес в интернете</h2>

<div class="domain-structure">
 www . moy-sait .
ru

 ▲ ▲ ▲

 поддомен
 название
 зона
</div>

<div class="note">
 💡 Запомните: Домен — это ВАШЕ уникальное имя в интернете. Как только Вы регистрируете домен, никто другой не может использовать это же имя.
</div>

<h3>Популярные доменные зоны:</h3>

<table class="comparison-table">
 <tr>
 <th>Доменная зона</th>
 <th>Для кого подходит</th>
 <th>Пример</th>
 <th>Особенности</th>
 </tr>
 <tr>
 <td>.ru</td>

```
<td>Для сайтов в России</td>
<td>yandex.ru</td>
<td>Популярен в РФ, недорогой</td>
</tr>
<tr>
    <td><strong>.com</strong></td>
    <td>Международные сайты</td>
    <td>google.com</td>
    <td>Самый узнаваемый в мире</td>
</tr>
<tr>
    <td><strong>.рф</strong></td>
    <td>Русскоязычные сайты</td>
    <td>почта.рф</td>
    <td>Кириллический домен</td>
</tr>
<tr>
    <td><strong>.net</strong></td>
    <td>Технические проекты</td>
    <td>habr.net</td>
    <td>Изначально для сетевых технологий</td>
</tr>
<tr>
    <td><strong>.org</strong></td>
    <td>Некоммерческие организации</td>
    <td>wikipedia.org</td>
    <td>Для благотворительности, сообществ</td>
</tr>
</table>
```

```
<div class="warning">
    <strong>⚠ Важно:</strong> Домен нужно ежегодно продлевать!
```

Это как аренда адреса.

Если не продлить вовремя, домен освободится и его сможет занять кто-то другой.

```
</div>
</div>
```

```
<!-- Подробно о хостинге -->
```

```
<div class="example-section">
```

```
    <h2>3. Что такое хостинг? Земля под Ваш сайт</h2>
```

```
<div class="note">
```

 Хостинг — это: Услуга по предоставлению места для Вашего сайта на специальном компьютере (сервере), который работает 24/7 и всегда подключен к интернету.

</div>

<h3>Типы хостинга (для новичков):</h3>

<div class="hosting-types">

<div class="hosting-card">

<h3> Виртуальный хостинг</h3>

<p>Для: Личных сайтов, блогов, небольших проектов</p>

<p>Аналогия: Комната в многоквартирном доме</p>

<p>Плюсы: Недорогой, простой в использовании</p>

<p>Минусы: Ограниченные ресурсы</p>

<p>Цена: от 100 руб/месяц</p>

</div>

<div class="hosting-card">

<h3> VPS хостинг</h3>

<p>Для: Средних сайтов, интернет-магазинов</p>

<p>Аналогия: Отдельная квартира</p>

<p>Плюсы: Больше свободы и ресурсов</p>

<p>Минусы: Сложнее в настройке, дороже</p>

<p>Цена: от 300 руб/месяц</p>

</div>

<div class="hosting-card">

<h3> Выделенный сервер</h3>

<p>Для: Крупных проектов, соцсетей</p>

<p>Аналогия: Частный дом</p>

<p>Плюсы: Максимальная производительность</p>

<p>Минусы: Дорогой, нужен

администратор</p>

<p>Цена: от 2000 руб/месяц</p>

</div>

</div>

<div class="success">

💡 Для начинающих: Начните с виртуального хостинга! Его достаточно для 95% личных сайтов, и он самый простой в использовании.

</div>
</div>

<!-- Как это работает вместе -->

<div class="example-section">
 <h2>4. Как домен и хостинг работают вместе</h2>

<div class="step-by-step">
 <h3>⌚ Процесс за 4 шага:</h3>

<div class="step">
 Шаг 1: Пользователь вводит в браузере Ваш домен (например, moy-sait.ru)</div>

<div class="step">
 Шаг 2: Браузер обращается к системе DNS (как телефонный справочник интернета)</div>

<div class="step">
 Шаг 3: DNS находит, на каком хостинге находится сайт с этим доменом</div>

<div class="step">
 Шаг 4: Пользователь попадает на Ваш хостинг и видит Ваш сайт</div>
</div>

<div class="code-example">
// Техническое объяснение (для любознательных):

1. Пользователь: browser → "moy-sait.ru"
 2. DNS-сервер: "moy-sait.ru" → 192.168.1.1 (IP хостинга)
 3. Браузер: 192.168.1.1 → "покажи index.html"
 4. Хостинг: отправляет файлы сайта пользователю
- </div>

<div class="visual-explanation">
 <div class="visual-card">

```
<div class="visual-icon"></div>
<h3>Пользователь</h3>
<p>Вводит домен в браузере</p>
</div>

<div class="visual-card">
  <div class="visual-icon"></div>
  <h3>DNS система</h3>
  <p>Находит где находится сайт</p>
</div>

<div class="visual-card">
  <div class="visual-icon"></div>
  <h3>Хостинг</h3>
  <p>Отправляет файлы сайта</p>
</div>

<div class="visual-card">
  <div class="visual-icon"></div>
  <h3>Браузер</h3>
  <p>Показывает сайт пользователю</p>
</div>
</div>

<!-- Практические советы -->
<div class="example-section">
  <h2>5. Практические советы для начинающих</h2>

  <div style="display: grid; grid-template-columns: repeat(auto-fit, minmax(300px, 1fr)); gap: 15px; margin: 20px 0;">
    <div style="border: 2px solid #3498db; padding: 15px; border-radius: 8px;">
      <h3 style="color: #3498db; margin-top: 0;"> Выбор
      домена</h3>
      <ul>
        <li><strong>Короткий и запоминающийся</strong></li>
        <li><strong>Без сложного написания</strong></li>
        <li><strong>Отражает тематику сайта</strong></li>
        <li><strong>Проверьте на доступность</strong></li>
      </ul>
    </div>
  </div>

  <div style="border: 2px solid #27ae60; padding: 15px; border-radius: 8px;">
```

```
<h3 style="color: #27ae60; margin-top: 0;">💻 Выбор
хостинга</h3>
<ul>
    <li><strong>Начните с виртуального</strong></li>
    <li><strong>Русскоязычная поддержка</strong></li>
    <li><strong>Простая панель управления</strong></li>
    <li><strong>Техподдержка 24/7</strong></li>
</ul>
</div>

<div style="border: 2px solid #e74c3c; padding: 15px; border-radius:
8px;">
    <h3 style="color: #e74c3c; margin-top: 0;">💰 Бюджет</h3>
    <ul>
        <li><strong>Домен:</strong> 500-1500 руб/год</li>
        <li><strong>Хостинг:</strong> 100-500 руб/месяц</li>
        <li><strong>Итого:</strong> ~2000 руб/год</li>
        <li><strong>Экономия:</strong> Бесплатные аналоги</li>
    </ul>
    </div>
</div>

<div class="warning">
    <strong>⚠ Чего избегать:</strong>
    <ul>
        <li><strong>Бесплатных хостингов</strong> с навязчивой
рекламой</li>
        <li><strong>Слишком дешёвых предложений</strong> ("10
руб/месяц")</li>
        <li><strong>Доменов в экзотических зонах</strong> (.tk, .ml и
т.д.)</li>
        <li><strong>Хостингов без техподдержки</strong> на русском
языке</li>
    </ul>
    </div>
</div>

<!-- Популярные провайдеры -->
<div class="example-section">
    <h2>6. Где зарегистрировать домен и купить хостинг?</h2>
    <h3>RU Популярные российские провайдеры:</h3>
    <table class="comparison-table">
        <tr>
```

Провайдер	Домены	Хостинг	Особенности	Для новичков
<td>Reg.ru</td>	<td>  </td>	<td>  </td>	<td>Крупнейший в РФ, много услуг</td>	<td>★ ★ ★ ★ ★</td>
<td>Nic.ru</td>	<td>  </td>	<td>  </td>	<td>Надёжный, официальный регистратор</td>	<td>★ ★ ★ ★ ★</td>
<td>Beget.com</td>	<td>  </td>	<td>  </td>	<td>Хорошая техподдержка</td>	<td>★ ★ ★ ★ ★</td>
<td>Timeweb.com</td>	<td>  </td>	<td>  </td>	<td>Удобная панель управления</td>	<td>★ ★ ★ ★ ★</td>
<td>SpaceWeb.com</td>	<td>  </td>	<td>  </td>	<td>Простой интерфейс</td>	<td>★ ★ ★ ★ ★</td>

<div class="note">

💡 Рекомендация: Для первого раза выберите провайдера, где можно и домен зарегистрировать, и хостинг купить в одном месте — так проще управлять.

</div>
</div>

<!-- Бесплатные альтернативы -->

<div class="example-section">

<h2>7. Бесплатные альтернативы для обучения</h2>

<div class="warning">

🎓 Для учёбы и практики можно использовать бесплатные варианты:

</div>

<div style="display: grid; grid-template-columns: repeat(auto-fit, minmax(250px, 1fr)); gap: 15px; margin: 20px 0;">

<div style="border: 2px solid #3498db; padding: 15px; border-radius: 8px; text-align: center;">

<h4>GitHub Pages</h4>

<p>Бесплатный хостинг для статических сайтов</p>

<p>Ваш домен: username.github.io</p>

<p>✓ Идеально для HTML/CSS</p>

</div>

<div style="border: 2px solid #27ae60; padding: 15px; border-radius: 8px; text-align: center;">

<h4>Netlify</h4>

<p>Бесплатный хостинг с простой загрузкой</p>

<p>Можно подключить свой домен</p>

<p>✓ Очень прост в использовании</p>

</div>

<div style="border: 2px solid #9b59b6; padding: 15px; border-radius: 8px; text-align: center;">

<h4>Бесплатные поддомены</h4>

<p>some-site.nethouse.ru</p>

<p>my-site.ucoz.net</p>

<p>✓ Для первых экспериментов</p>

</div>

</div>

<div class="note">

 Для серьёзных проектов лучше использовать платный хостинг и домен — это надёжнее и даёт больше возможностей для роста.

</div>
</div>

<!-- Практическое задание -->

<div class="example-section">
 <h2> Практическое задание: Спланируй свой первый сайт</h2>

<div class="practice-task">
 <h3>Подготовься к созданию настоящего сайта в интернете!</h3>

<div class="interactive-demo">
 <h4> Задание:</h4>

<div style="background: #f5f5f5; padding: 20px; border-radius: 8px; margin: 15px 0;">
 1. Придумай название для сайта:

 <input type="text" placeholder="Например: мой-садовый-дневник" style="width: 100%; padding: 10px; margin: 10px 0; border: 1px solid #ddd; border-radius: 5px;">

2. Выбери доменную зону:

 <select style="width: 100%; padding: 10px; margin: 10px 0; border: 1px solid #ddd; border-radius: 5px;">
 <option>.ru (для России)</option>
 <option>.com (международный)</option>
 <option>.рф (русский язык)</option>
 <option>.net (технический)</option>
 </select>

3. Твой будущий домен:

 <div style="background: white; padding: 15px; border: 2px dashed #3498db; border-radius: 5px; text-align: center; font-size: 1.2em; margin: 10px 0;">
 мой-сайт.ru
 </div>

4. Бюджет на год:

 <div style="background: white; padding: 15px; border-radius: 5px; margin: 10px 0;">
 Домен: ~800 руб/год + Хостинг: ~1500 руб/год =
 ~2300 руб/год
 </div>

```
</div>
</div>

<div class="success">
    <strong>★ Поздравляю!</strong> Теперь ты понимаешь, что
нужно для создания настоящего сайта в интернете.
```

В следующих уроках мы научимся загружать твой HTML-код на хостинг!

```
</div>
</div>
</div>
```

```
<script>
    // Обновление предпросмотра домена
    const nameInput = document.querySelector('input[type="text"]');
    const zoneSelect = document.querySelector('select');
    const domainPreview = document.getElementById('domainPreview');

    function updateDomainPreview() {
        const name = nameInput.value || 'мой-сайт';
        const zone = zoneSelect.value.split(' ')[0];
        domainPreview.textContent = name + zone;
    }

    nameInput.addEventListener('input', updateDomainPreview);
    zoneSelect.addEventListener('change', updateDomainPreview);
    updateDomainPreview(); // Инициализация
</script>
</body>
</html>
```

Часть 7: Что Вы теперь знаете

- Понимаете аналогию домена и хостинга с реальным миром
- Знаете, что такое домен и из чего он состоит
- Понимаете, что такое хостинг и какие типы бывают
- Видите, как домен и хостинг работают вместе
- Можете выбрать подходящий домен и хостинг для своего проекта
- Знаете популярных провайдеров на российском рынке
- Осознаёте бюджет для создания своего сайта

Часть 8: Итоги и рекомендации

-  Ключевые выводы:

Домен и хостинг — это основа любого сайта в интернете
Домен = адрес, хостинг = земля под строительство
Для начинающих лучше начинать с виртуального хостинга
Домены нужно ежегодно продлевать — это аренда, а не покупка
Российские провайдеры часто удобнее для новичков из-за русской поддержки

Практические рекомендации для первого сайта:

Начните с .ru домена — привычно для русскоязычных пользователей
Выберите виртуальный хостинг — его хватит с запасом для личного сайта
Ищите пакетные предложения — домен + хостинг в одном месте
Обращайте внимание на поддержку — она понадобится в первые дни
Не экономьте на качестве — дешёвый хостинг может создать больше проблем, чем решить

Следующие шаги:

Придумайте название для своего сайта
Проверьте доступность домена на сайтах регистраторов
Выберите хостинг-провайдера из рекомендованных
Подготовьте бюджет на первый год (2000-3000 рублей)
В следующем уроке научимся загружать файлы на хостинг!

Помните: Регистрация домена и хостинга — это первый шаг к тому, чтобы Ваш сайт увидел весь мир! 

Теперь Вы вооружены знаниями, чтобы выбрать себе красивый адрес в интернете и надёжное место для Вашего будущего сайта! 

13.2. Обзор бесплатных хостингов (например, GitHub Pages — объяснение максимально просто)

Введение: Ваш сайт бесплатно в интернете!

Представьте, что Вам подарили бесплатную квартиру в интернете для Вашего сайта! Бесплатные хостинги — это именно такие "подарочные квартиры", где Вы можете разместить свой сайт без оплаты. Сегодня мы разберём самый популярный бесплатный вариант — GitHub Pages.

Часть 1: Что такое GitHub Pages? Очень простое объяснение

1.1. GitHub Pages — это как бесплатная витрина для ваших проектов

html

```
<div class="simple-explanation">
  <h3>🎁 GitHub Pages — это:</h3>
  <div class="analogy">
    <strong>Реальный мир:</strong> Бесплатная витрина в торговом
    центре<br>
    <strong>Интернет:</strong> Бесплатное место для Вашего
    сайта<br>
    <strong>Что нужно:</strong> Только Ваши HTML/CSS файлы
  </div>
</div>
```

1.2. Как это работает — простая схема:

text



Часть 2: Почему именно GitHub Pages? 5 главных преимуществ

html

```
<div class="advantages">
  <div class="advantage">
    <strong>1.  Бесплатно</strong> — Никакой оплаты, всегда
    бесплатно
  </div>
  <div class="advantage">
    <strong>2.  Просто</strong> — Загрузил файлы и сайт работает
  </div>
  <div class="advantage">
    <strong>3.  Надёжно</strong> — От компании Microsoft
  </div>
  <div class="advantage">
    <strong>4.  Авто-обновления</strong> — Обновляется
    автоматически
  </div>
  <div class="advantage">
    <strong>5.  Доступно сразу</strong> — Сразу по адресу
    username.github.io
  </div>
</div>
```

Часть 3: Полная страница с пошаговым объяснением

html

```
<!DOCTYPE html>
<html lang="ru">
<head>
  <meta charset="utf-8">
  <title>Бесплатные хостинги: GitHub Pages для начинающих</title>
  <style>
    body {
      font-family: 'Arial', sans-serif;
      max-width: 1200px;
      margin: 0 auto;
      padding: 20px;
      line-height: 1.6;
      background-color: #f8f9fa;
    }
    .example-section {
      background: white;
      padding: 25px;
      margin: 25px 0;
    }
  </style>
</head>
<body>
  <h1>Бесплатные хостинги: GitHub Pages для начинающих</h1>
  <h2>Шаг 1: Создание репозитория на GitHub</h2>
  <p>Создайте новый репозиторий на GitHub, выбрав опцию "Initialize this repository with a README".</p>
  <h2>Шаг 2: Клонирование репозитория на локальную машину</h2>
  <p>Клонируйте репозиторий с помощью команды git clone https://github.com/username/repo.git в директорию, где вы хотите разместить сайт.</p>
  <h2>Шаг 3: Создание файла README.md</h2>
  <p>Создайте файл README.md в корневой директории вашего проекта. Это будет вашим первым страницей сайта.</p>
  <h2>Шаг 4: Настройка файла .gitignore</h2>
  <p>Создайте файл .gitignore в корневой директории. Установите его содержимое так, чтобы исключить из отслеживания файлы, которые не должны быть размещены в репозитории (например, временные файлы).</p>
  <h2>Шаг 5: Настройка файла _config.yml</h2>
  <p>Создайте файл _config.yml в корневой директории. В нем укажите параметры для GitHub Pages, такие как язык и тема. Для начала можно использовать предустановленные темы GitHub Pages.</p>
  <h2>Шаг 6: Публикация сайта</h2>
  <p>После выполнения всех шагов, нажмите кнопку "Commit changes" на GitHub и выберите ветку "gh-pages". Ваш сайт будет автоматически опубликован на GitHub Pages под адресом https://username.github.io/repo/.</p>
</body>
</html>
```

```
border-radius: 10px;
box-shadow: 0 2px 5px rgba(0,0,0,0.1);
}
h1, h2 {
    color: #2c3e50;
}
h1 {
    border-bottom: 3px solid #3498db;
    padding-bottom: 15px;
}
.code-example {
    background: #2d2d2d;
    color: #f8f8f2;
    padding: 15px;
    border-radius: 5px;
    font-family: 'Courier New', monospace;
    margin: 15px 0;
    overflow-x: auto;
}
.note {
    background: #e7f3ff;
    padding: 15px;
    border-left: 4px solid #2196F3;
    margin: 15px 0;
    border-radius: 5px;
}
.warning {
    background: #fff3e0;
    padding: 15px;
    border-left: 4px solid #ff9800;
    margin: 15px 0;
    border-radius: 5px;
}
.success {
    background: #e8f5e8;
    padding: 15px;
    border-left: 4px solid #4CAF50;
    margin: 15px 0;
    border-radius: 5px;
}
.analogy-box {
    background: #f3e5f5;
    border: 2px dashed #9c27b0;
    border-radius: 8px;
    padding: 20px;
```

```
        margin: 20px 0;
    }
.step-by-step {
    background: #fff3e0;
    padding: 20px;
    border-radius: 8px;
    margin: 20px 0;
}
.step {
    margin: 20px 0;
    padding: 15px;
    background: white;
    border-radius: 8px;
    border-left: 4px solid #3498db;
}
.step-number {
    background: #3498db;
    color: white;
    width: 30px;
    height: 30px;
    border-radius: 50%;
    display: inline-flex;
    align-items: center;
    justify-content: center;
    margin-right: 10px;
    font-weight: bold;
}
.visual-guide {
    display: grid;
    grid-template-columns: repeat(auto-fit, minmax(300px, 1fr));
    gap: 20px;
    margin: 20px 0;
}
.visual-card {
    background: white;
    border: 2px solid #3498db;
    border-radius: 10px;
    padding: 20px;
    text-align: center;
}
.visual-icon {
    font-size: 3em;
    margin-bottom: 15px;
}
.comparison-table {
```

```
width: 100%;  
border-collapse: collapse;  
margin: 20px 0;  
}  
.comparison-table th, .comparison-table td {  
border: 1px solid #ddd;  
padding: 12px;  
text-align: left;  
}  
.comparison-table th {  
background: #34495e;  
color: white;  
}  
.comparison-table tr:nth-child(even) {  
background: #f9f9f9;  
}  
.free-options {  
display: grid;  
grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));  
gap: 15px;  
margin: 20px 0;  
}  
.option-card {  
background: white;  
border: 1px solid #e0e0e0;  
border-radius: 8px;  
padding: 20px;  
}  
.option-card h3 {  
color: #2c3e50;  
margin-top: 0;  
display: flex;  
align-items: center;  
gap: 10px;  
}  
.interactive-demo {  
background: #e3f2fd;  
padding: 20px;  
border-radius: 8px;  
margin: 20px 0;  
}  
.github-preview {  
background: #24292e;  
color: white;  
padding: 15px;
```

```
border-radius: 5px;  
font-family: 'Courier New', monospace;  
margin: 15px 0;  
text-align: center;  
}  
</style>  
</head>  
<body>  
  <h1>🎁 Бесплатные хостинги: Размещаем сайт бесплатно с GitHub  
Pages</h1>  
  
<div class="note">  
  <strong>💡 Основная идея:</strong> Не обязательно платить за  
хостинг, чтобы разместить свой сайт в интернете! Существуют  
бесплатные варианты, которые идеально подходят для обучения и  
личных проектов.  
</div>  
  
<!-- Что такое GitHub Pages -->  
<div class="example-section">  
  <h2>1. Что такое GitHub Pages? Объясняем на пальцах</h2>  
  
  <div class="analogy-box">  
    <h3>🏢 Простая аналогия:</h3>  
    <p><strong>GitHub Pages</strong> — это как бесплатная витрина  
в большом торговом центре (GitHub).</p>  
    <p>Вы получаете:</p>  
    <ul>  
      <li>✓ <strong>Бесплатное место</strong> для своего  
сайта</li>  
      <li>✓ <strong>Красивый адрес</strong> ваш-  
логин.github.io</li>  
      <li>✓ <strong>Автоматическое обслуживание</strong> — всё  
работает само</li>  
      <li>✓ <strong>Защиту и надёжность</strong> от большой  
компании</li>  
    </ul>  
  </div>  
  
<div class="visual-guide">  
  <div class="visual-card">  
    <div class="visual-icon">📁</div>  
    <h3>Ваши файлы</h3>  
    <p>HTML, CSS, JavaScript файлы которые Вы создали</p>  
  </div>
```

```
<div class="visual-card">
  <div class="visual-icon"> </div>
  <h3>Загружаете</h3>
  <p>Помещаете файлы в специальное хранилище</p>
</div>

<div class="visual-card">
  <div class="visual-icon"> </div>
  <h3>Получаете сайт</h3>
  <p>Ваш-логин.github.io — готовый сайт!</p>
</div>
</div>

<div class="github-preview">
  https:// ваш-логин .github.io
</div>
</div>

<!-- Преимущества GitHub Pages --&gt;
&lt;div class="example-section"&gt;
  &lt;h2&gt;2. Почему GitHub Pages идеален для начинающих?&lt;/h2&gt;

  &lt;table class="comparison-table"&gt;
    &lt;tr&gt;
      &lt;th&gt;Преимущество&lt;/th&gt;
      &lt;th&gt;Объяснение&lt;/th&gt;
      &lt;th&gt;Почему это важно&lt;/th&gt;
    &lt;/tr&gt;
    &lt;tr&gt;
      &lt;td&gt;&lt;strong&gt;<img alt="Clock icon" data-bbox="345 635 365 655"/> Полностью бесплатно</strong></td>
      <td>Никаких скрытых платежей или ограничений по времени</td>
      <td>Можно учиться и экспериментировать без риска</td>
    </tr>
    <tr>
      <td><strong> Простота использования</strong></td>
      <td>Просто загрузил файлы — сайт работает</td>
      <td>Не нужно разбираться в сложных настройках</td>
    </tr>
    <tr>
      <td><strong> Надёжность</strong></td>
      <td>От компании Microsoft, работает 24/7</td>
      <td>Ваш сайт всегда доступен</td>
    </tr>
  </table>
</div>
```

```

<tr>
    <td><strong>💡 Автоматическое обновление</strong></td>
    <td>Обновил файлы — сайт сразу меняется</td>
    <td>Не нужно ничего перезагружать вручную</td>
</tr>
<tr>
    <td><strong>🌐 Готовый домен</strong></td>
    <td>Сразу получаете красивый адрес</td>
    <td>Не нужно покупать и настраивать домен</td>
</tr>
</table>

<div class="success">
    <strong>💡 Идеально для:</strong> Личных сайтов, портфолио, учебных проектов, блогов, сайтов-визиток, демонстрационных страниц.
</div>
</div>

<!-- Пошаговая инструкция -->
<div class="example-section">
    <h2>3. Пошаговая инструкция: Как создать сайт на GitHub Pages</h2>

    <div class="step-by-step">
        <div class="step">
            <div class="step-number">1</div>
            <h3>Создайте аккаунт на GitHub</h3>
            <p><strong>Что делать:</strong> Перейдите на <a href="https://github.com" target="_blank">github.com</a> и зарегистрируйтесь</p>
            <p><strong>Просто:</strong> Как регистрация в любой социальной сети</p>
            <div class="code-example">
                // Адрес для регистрации:  

                https://github.com/signup

                // Что нужно:
                • Придумать логин (это будет часть Вашего домена)
                • Указать email
                • Придумать пароль
                    </div>
                </div>

                <div class="step">
                    <div class="step-number">2</div>

```

```
<h3>Создайте новый репозиторий</h3>
<p><strong>Что такое репозиторий:</strong> Это как папка-
хранилище для ваших файлов</p>
<p><strong>Важно:</strong> Название должно быть
<strong> ваш-логин.github.io</strong></p>
<div class="code-example">
// Пример для пользователя ivanov:
Название репозитория: ivanov.github.io

// Обязательно:
• Public (публичный)
• Initialize with README (поставить галочку)
</div>
</div>

<div class="step">
<div class="step-number">3</div>
<h3>Загрузите Ваши HTML файлы</h3>
<p><strong>Что загружать:</strong> Все файлы Вашего
сайта</p>
<p><strong>Обязательно:</strong> Главная страница должна
называться <strong>index.html</strong></p>
<div class="code-example">
// Структура файлов:
ivanov.github.io/
└── index.html      (главная страница)
└── style.css       (стили)
└── about.html      (другие страницы)
└── images/
    └── photo.jpg
</div>
</div>

<div class="step">
<div class="step-number">4</div>
<h3>Включите GitHub Pages</h3>
<p><strong>Где найти:</strong> Настройки репозитория →
раздел Pages</p>
<p><strong>Что сделать:</strong> Выбрать ветку "main" и
нажать Save</p>
<div class="code-example">
// Путь в интерфейсе:
Your Repository → Settings → Pages
→ Source: Deploy from a branch
→ Branch: main
```

→ Save

```
</div>
</div>

<div class="step">
  <div class="step-number">5</div>
  <h3>Готово! Ваш сайт работает</h3>
  <p><strong>Адрес сайта:</strong> https://ваш-
логин.github.io</p>
  <p><strong>Проверьте:</strong> Откройте этот адрес в
браузере</p>
  <div class="github-preview">
    https://<strong>ваш-логин</strong>.github.io
  </div>
</div>
</div>
</div>

<!-- Другие бесплатные варианты -->
<div class="example-section">
  <h2>4. Другие бесплатные хостинги — обзор альтернатив</h2>

  <div class="free-options">
    <div class="option-card">
      <h3> Netlify</h3>
      <p><strong>Простота:</strong> Очень простой, перетащи
файлы</p>
      <p><strong>Домен:</strong> случайный.netlify.app</p>
      <p><strong>Плюсы:</strong> Супер-простой интерфейс</p>
      <p><strong>Для кого:</strong> Для абсолютных новичков</p>
    </div>
    <div class="option-card">
      <h3> Vercel</h3>
      <p><strong>Простота:</strong> Простой, похож на Netlify</p>
      <p><strong>Домен:</strong> случайный.vercel.app</p>
      <p><strong>Плюсы:</strong> Хорошая скорость</p>
      <p><strong>Для кого:</strong> Для современных проектов</p>
    </div>
    <div class="option-card">
      <h3> GitLab Pages</h3>
      <p><strong>Простота:</strong> Похож на GitHub Pages</p>
      <p><strong>Домен:</strong> ваш-логин.gitlab.io</p>
    </div>
  </div>
</div>
```

<p>Плюсы: Больше бесплатных возможностей</p>

<p>Для кого: Для тех, кто хочет альтернативу</p>

</div>

<div class="option-card">

<h3> Firebase Hosting</h3>

<p>Простота: Средняя, нужно установить программу</p>

<p>Домен: случайный.web.app</p>

<p>Плюсы: От Google, много дополнительных функций</p>

<p>Для кого: Для более продвинутых пользователей</p>

</div>

</div>

<div class="note">

 Рекомендация для начинающих: Начните с GitHub Pages или Netlify.

Они самые простые в использовании и не требуют специальных знаний.

</div>

</div>

<!-- Ограничения бесплатных хостингов -->

<div class="example-section">

<h2>5. Что МОЖНО и что НЕЛЬЗЯ на бесплатных хостингах</h2>

<div style="display: grid; grid-template-columns: 1fr 1fr; gap: 20px; margin: 20px 0;">

<div style="background: #e8f5e8; padding: 20px; border-radius: 8px;">

<h3 style="color: #27ae60; margin-top: 0;"> ЧТО МОЖНО</h3>

Статические сайты (HTML, CSS, JavaScript)

Личные блоги и портфолио

Учебные проекты и демо

Сайты-визитки и презентации

Документацию и инструкции

Онлайн-резюме и портфолио

</div>

```
<div style="background: #ffebee; padding: 20px; border-radius: 8px;">
  <h3 style="color: #e74c3c; margin-top: 0;">✖ ЧТО НЕЛЬЗЯ</h3>
  <ul>
    <li><strong>Интернет-магазины</strong> с оплатой</li>
    <li><strong>Сайты с базой данных</strong> (PHP +
      MySQL)</li>
    <li><strong>Социальные сети</strong> и форумы</li>
    <li><strong>Сайты для бизнеса</strong> с высокой
      нагрузкой</li>
    <li><strong>Платформы для видео-хостинга</strong></li>
    <li><strong>Сайты с бэкенд-логикой</strong></li>
  </ul>
</div>
</div>
```

```
<div class="warning">
  <strong>⚠ Важное ограничение:</strong> Бесплатные хостинги
  подходят только для <strong>статических сайтов</strong>. Это сайты,
  которые показывают готовые HTML-страницы без обработки на сервере.
</div>
</div>
```

<!-- Сравнение с платным хостингом -->

```
<div class="example-section">
  <h2>6. Сравнение: Бесплатный vs Платный хостинг</h2>
```

Параметр	Бесплатный хостинг	Платный хостинг	Что выбрать?
Стоимость	 Бесплатно	от 100 руб/месяц	Бесплатный для учёбы
Домен	ваш-логин.github.io	Любой домен (мой-сайт.ru)	Бесплатный для тестов

```
</tr>
<tr>
    <td><strong>Поддержка</strong></td>
    <td>Форум, документация</td>
    <td>Техподдержка 24/7</td>
    <td>Платный для бизнеса</td>
</tr>
<tr>
    <td><strong>Безопасность</strong></td>
    <td>Хорошая</td>
    <td>Высокая + SSL</td>
    <td>Оба варианта надёжны</td>
</tr>
<tr>
    <td><strong>Функции</strong></td>
    <td>Только статические сайты</td>
    <td>PHP, базы данных, почта</td>
    <td>Платный для сложных проектов</td>
</tr>
<tr>
    <td><strong>Простота</strong></td>
    <td>✓ Очень просто</td>
    <td>⚠ Нужно разбираться</td>
    <td>Бесплатный для начинающих</td>
</tr>
</table>
```

```
<div class="note">
```

🎓 Для обучения: Начинайте с бесплатного хостинга. Когда Ваш проект вырастет и начнёт приносить доход, можно перейти на платный хостинг.

```
</div>
```

```
</div>
```

```
<!-- Практическое задание -->
```

```
<div class="example-section">
```

<h2>🎯 Практическое задание: Создай свой первый сайт на GitHub Pages!</h2>

```
<div class="practice-task">
```

<h3>Размести свой HTML-проект в интернете бесплатно</h3>

```
<div class="interactive-demo">
```

<h4>📋 Пошаговый план:</h4>

```
<div style="background: #f5f5f5; padding: 20px; border-radius: 8px; margin: 15px 0;">
    <div class="step">
        <div class="step-number">1</div>
        <strong>Подготовь файлы сайта:</strong><br>
        <textarea placeholder="Убедись, что у тебя есть:
            • index.html (главная страница)
            • style.css (стили)
            • images/ (папка с картинками)" style="width: 100%; height: 100px; padding: 10px; margin: 10px 0; border: 1px solid #ddd; border-radius: 5px;"></textarea>
    </div>

    <div class="step">
        <div class="step-number">2</div>
        <strong>Зарегистрируйся на GitHub:</strong><br>
        <input type="text" placeholder="Придумай логин (это будет твой домен)" style="width: 100%; padding: 10px; margin: 10px 0; border: 1px solid #ddd; border-radius: 5px;">
    </div>

    <div class="step">
        <div class="step-number">3</div>
        <strong>Твой будущий сайт будет доступен по адресу:</strong><br>
        <div class="github-preview">
            https://<span id="usernamePreview">твой-логин</span>.github.io
        </div>
    </div>
</div>

<div class="success">
    <strong>🏆 Цель задания:</strong> Создать аккаунт на GitHub и подготовить файлы для загрузки.
    В следующем уроке мы подробно разберём процесс загрузки файлов на GitHub Pages!
</div>
</div>
</div>
</div>

<!-- Частые вопросы -->
<div class="example-section">
    <h2>7. Ответы на частые вопросы (FAQ)</h2>
```

```
<div style="background: #f8f9fa; padding: 20px; border-radius: 8px;  
margin: 20px 0;">  
    <h4>❓ Нужно ли знать программирование для использования  
GitHub Pages?</h4>  
    <p><strong>Ответ:</strong> НЕТ! Достаточно уметь создавать  
HTML и CSS файлы. Интерфейс GitHub очень простой.</p>  
  
    <h4>❓ Можно ли поменять дизайн сайта после размещения?  
</h4>  
    <p><strong>Ответ:</strong> ДА! Просто загрузите новые версии  
файлов, и сайт автоматически обновится.</p>  
  
    <h4>❓ Будет ли реклама на моём сайте?</h4>  
    <p><strong>Ответ:</strong> НЕТ! GitHub Pages не добавляет  
никакой рекламы на Ваши сайты.</p>  
  
    <h4>❓ Сколько сайтов можно создать бесплатно?</h4>  
    <p><strong>Ответ:</strong> Неограниченное количество! Один  
основной сайт и много проектных страниц.</p>  
  
    <h4>❓ Что делать, если я ошибся при загрузке файлов?</h4>  
    <p><strong>Ответ:</strong> Можно просто удалить неправильные  
файлы и загрузить заново. Это безопасно.</p>  
    </div>  
</div>  
  
<script>  
// Обновление предпросмотра домена GitHub  
const usernameInput = document.querySelector('input[type="text"]');  
const usernamePreview =  
document.getElementById('usernamePreview');  
  
usernameInput.addEventListener('input', function() {  
    usernamePreview.textContent = this.value || 'твой-логин';  
});  
</script>  
</body>  
</html>
```

Часть 7: Что Вы теперь знаете о бесплатных хостингах

- Понимаете, что такое GitHub Pages и как он работает
- Знаете преимущества бесплатных хостингов для обучения
- Видите пошаговый процесс создания сайта на GitHub Pages

- Знаете альтернативные бесплатные хостинги
- Понимаете ограничения бесплатных решений
- Можете сравнить бесплатные и платные хостинги
- Готовы создать свой первый сайт на GitHub Pages

Часть 8: Итоги и рекомендации

Ключевые выводы о бесплатных хостингах:

GitHub Pages — идеальный старт для начинающих
Всё действительно бесплатно — никаких скрытых платежей
Процесс очень простой — не требует специальных знаний
Надёжность обеспечена компанией Microsoft
Идеально для статических сайтов — HTML, CSS, JavaScript

Практические рекомендации:

Начните с GitHub Pages — это самый популярный и документированный вариант
Используйте для обучения — можно делать сколько угодно учебных проектов
Создайте портфолио — прекрасное место для демонстрации ваших работ
Экспериментируйте смело — ничего не сломается, всё можно исправить

Почему стоит начать именно с GitHub Pages:

Бесплатно навсегда — не нужно беспокоиться об оплате
Профessionальная среда — этот навык пригодится в будущем
Интеграция с Git — познакомитесь с системой контроля версий
Большое сообщество — много учебных пособий и помощи
Масштабируемость — можно начать с простого и развивать проект

Когда стоит переходить на платный хостинг:

- Когда сайт начинает приносить доход
- Когда нужны динамические функции (базы данных, PHP)
- Когда нужен красивый собственный домен
- Когда проект требует технической поддержки
- Когда трафик становится очень большим

Помните: Бесплатные хостинги — это прекрасная возможность научиться и создать свои первые проекты без финансовых рисков!

Теперь у Вас есть все знания, чтобы бесплатно разместить Ваш сайт в интернете и показать его всему миру! 

13.3. Простая инструкция: как загрузить файлы своего сайта в Интернет

Введение: Ваш сайт выходит в свет!

Представьте, что Вы написали книгу и теперь отправляете её в печать, чтобы все могли её прочитать. Загрузка файлов сайта в интернет — это такой же волнующий момент! Сегодня мы step-by-step разберём, как разместить Ваш HTML-сайт в интернете бесплатно.

Часть 1: Подготовка файлов — что нужно загружать

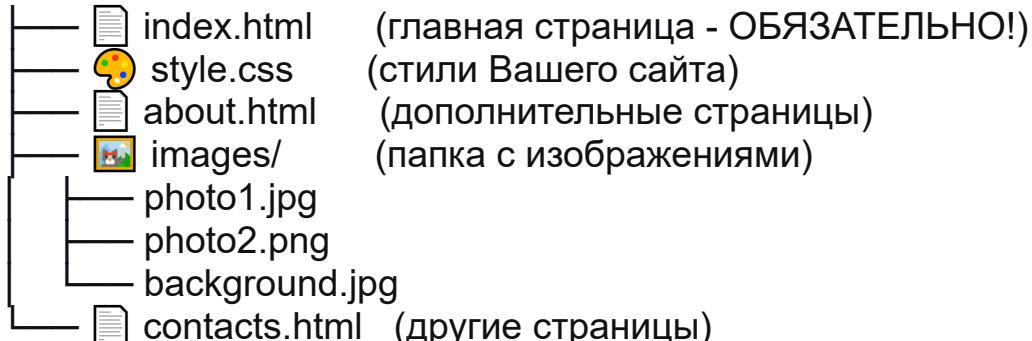
1.1. Проверяем структуру файлов

Перед загрузкой убедитесь, что у Вас есть все необходимые файлы:

html

<!-- Правильная структура папки с сайтом -->

мои-сайт/



! Важно: Главная страница ДОЛЖНА называться **index.html** — это стандарт для всех хостингов.

Часть 2: Выбор способа загрузки — 3 простых варианта

2.1. Вариант 1: GitHub Pages (рекомендуется)



Бесплатно

↗ Просто

🔒 Надёжно

2.2. Вариант 2: Netlify (очень просто)



Бесплатно

🖱️ Drag-and-drop (перетащил файлы — готово)

↗ Автоматическая настройка

2.3. Вариант 3: Бесплатный хостинг (для первых экспериментов)



Бесплатно



Обычно с рекламой



Только для обучения

Часть 3: Полная пошаговая инструкция для GitHub Pages

html

```
<!DOCTYPE html>
<html lang="ru">
<head>
  <meta charset="utf-8">
  <title>Простая инструкция: Как загрузить сайт в интернет</title>
  <style>
    body {
      font-family: 'Arial', sans-serif;
      max-width: 1200px;
      margin: 0 auto;
      padding: 20px;
      line-height: 1.6;
      background-color: #f8f9fa;
    }
    .example-section {
      background: white;
      padding: 25px;
      margin: 25px 0;
      border-radius: 10px;
      box-shadow: 0 2px 5px rgba(0,0,0,0.1);
    }
    h1, h2 {
      color: #2c3e50;
    }
    h1 {
      border-bottom: 3px solid #3498db;
      padding-bottom: 15px;
    }
    .code-example {
      background: #2d2d2d;
      color: #f8f8f2;
      padding: 15px;
      border-radius: 5px;
      font-family: 'Courier New', monospace;
      margin: 15px 0;
    }
  </style>
</head>
<body>
  <div class="example-section">
    <h1>Простая инструкция: Как загрузить сайт в интернет</h1>
    <h2>Шаг 1: Создание сайта</h2>
    <p>Создайте новый проект на GitHub Pages. Выберите шаблон и назовите его, например, "MyFirstSite".</p>
    <h2>Шаг 2: Добавление содержимого</h2>
    <p>Добавьте файл index.html в корень вашего проекта. Внутри файла напишите простую страницу с текстом:</p>
    <pre><code><h1>Привет, мир!</h1></code></pre>
    <h2>Шаг 3: Публикация сайта</h2>
    <p>После того как вы добавили содержимое, перейдите в раздел "Settings" (Настройки) вашего проекта на GitHub Pages. Выберите "Deploy to GitHub Pages" и выберите ветвь, на которую вы хотите развернуть сайт. Нажмите "Deploy branch" (Развернуть ветвь).</p>
    <h2>Шаг 4: Установка доменного имени</h2>
    <p>Если у вас есть собственное доменное имя, вы можете его настроить в разделе "Custom Domains" (Кастомные домены) настройках GitHub Pages. Установите доменное имя и выберите "Use GitHub Pages" (Использовать GitHub Pages).</p>
  </div>
</body>
</html>
```

```
    overflow-x: auto;
}
.note {
    background: #e7f3ff;
    padding: 15px;
    border-left: 4px solid #2196F3;
    margin: 15px 0;
    border-radius: 5px;
}
.warning {
    background: #fff3e0;
    padding: 15px;
    border-left: 4px solid #ff9800;
    margin: 15px 0;
    border-radius: 5px;
}
.success {
    background: #e8f5e8;
    padding: 15px;
    border-left: 4px solid #4CAF50;
    margin: 15px 0;
    border-radius: 5px;
}
.step-container {
    background: #fff3e0;
    padding: 20px;
    border-radius: 8px;
    margin: 20px 0;
}
.step {
    margin: 25px 0;
    padding: 20px;
    background: white;
    border-radius: 8px;
    border-left: 4px solid #3498db;
}
.step-number {
    background: #3498db;
    color: white;
    width: 35px;
    height: 35px;
    border-radius: 50%;
    display: inline-flex;
    align-items: center;
    justify-content: center;
```

```
margin-right: 15px;
font-weight: bold;
font-size: 1.2em;
}
.screenshot {
background: #2c3e50;
color: white;
padding: 15px;
border-radius: 5px;
margin: 15px 0;
font-family: 'Courier New', monospace;
}
.file-structure {
background: #e3f2fd;
padding: 15px;
border-radius: 5px;
margin: 15px 0;
font-family: 'Courier New', monospace;
}
.action-button {
background: #3498db;
color: white;
padding: 10px 20px;
border: none;
border-radius: 5px;
cursor: pointer;
font-size: 16px;
margin: 10px 5px;
}
.action-button:hover {
background: #2980b9;
}
.visual-guide {
display: grid;
grid-template-columns: repeat(auto-fit, minmax(300px, 1fr));
gap: 20px;
margin: 20px 0;
}
.visual-card {
background: white;
border: 2px solid #3498db;
border-radius: 10px;
padding: 20px;
text-align: center;
}
```

```
.visual-icon {  
    font-size: 3em;  
    margin-bottom: 15px;  
}  
.interactive-demo {  
    background: #e3f2fd;  
    padding: 20px;  
    border-radius: 8px;  
    margin: 20px 0;  
}  
.github-preview {  
    background: #24292e;  
    color: white;  
    padding: 15px;  
    border-radius: 5px;  
    font-family: 'Courier New', monospace;  
    margin: 15px 0;  
    text-align: center;  
    font-size: 1.2em;  
}  
.upload-methods {  
    display: grid;  
    grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));  
    gap: 15px;  
    margin: 20px 0;  
}  
.method-card {  
    background: white;  
    border: 1px solid #e0e0e0;  
    border-radius: 8px;  
    padding: 20px;  
}  
.method-card h3 {  
    color: #2c3e50;  
    margin-top: 0;  
    display: flex;  
    align-items: center;  
    gap: 10px;  
}  
}  
</style>  
</head>  
<body>  
    <h1> Простая инструкция: Как загрузить файлы своего сайта в  
Интернет</h1>
```

```
<div class="note">
    <strong>💡 Основная идея:</strong> Загрузка сайта в интернет —
это как отправить письмо по почте. Вы готовите конверт (файлы),
указываете адрес (домен) и отправляете на почту (хостинг). Сегодня мы
сделаем это вместе!
</div>
```

```
<!-- Подготовка файлов -->
<div class="example-section">
    <h2>1. Подготовка: Проверяем файлы перед загрузкой</h2>

    <div class="file-structure">
        <strong>📁 Ваша папка с сайтом должна содержать:</strong><br>
        мои-сайт/<br>
        └── <span style="color: #e74c3c;">index.html</span> ←
        ОБЯЗАТЕЛЬНО!<br>
        ├── style.css<br>
        ├── about.html<br>
        ├── images/<br>
        │   ├── photo1.jpg<br>
        │   └── logo.png<br>
        └── contacts.html
    </div>

    <div class="warning">
        <strong>❗ Проверьте перед загрузкой:</strong>
        <ul>
            <li>✓ Главный файл называется
        <strong>index.html</strong></li>
            <li>✓ Все картинки лежат в папке
        <strong>images/</strong></li>
            <li>✓ CSS файлы подключены правильно</li>
            <li>✓ Все ссылки между страницами работают</li>
            <li>✓ Проверили сайт на компьютере (открыли index.html в
браузере)</li>
        </ul>
    </div>

    <div class="interactive-demo">
        <h3>🎯 Быстрая проверка:</h3>
        <p>Откройте файл <strong>index.html</strong> в браузере прямо
с компьютера. Если всё отображается правильно — можно загружать!
</p>
    </div>

```

```
<button class="action-button" onclick="alert('Отлично! Откройте папку с Вашим сайтом и дважды щелкните на index.html')">Как проверить сайт</button>
</div>
</div>
```

```
<!-- Выбор способа загрузки -->
<div class="example-section">
    <h2>2. Выбираем способ загрузки: 3 простых варианта</h2>
```

```
<div class="upload-methods">
    <div class="method-card">
        <h3> GitHub Pages (рекомендуем)</h3>
        <p><strong>Сложность:</strong>  Легко</p>
        <p><strong>Стоимость:</strong>  Бесплатно</p>
        <p><strong>Домен:</strong> ваш-логин.github.io</p>
        <p><strong>Плюсы:</strong> Надёжно, профессионально, бесплатно</p>
        <button class="action-button"
            onclick="showGitHubSteps()">Выбрать этот способ</button>
    </div>

    <div class="method-card">
        <h3> Netlify Drop</h3>
        <p><strong>Сложность:</strong>  Очень легко</p>
        <p><strong>Стоимость:</strong>  Бесплатно</p>
        <p><strong>Домен:</strong> случайный.netlify.app</p>
        <p><strong>Плюсы:</strong> Просто перетащить файлы</p>
        <button class="action-button"
            onclick="showNetlifySteps()">Выбрать этот способ</button>
    </div>

    <div class="method-card">
        <h3> FileZilla + Хостинг</h3>
        <p><strong>Сложность:</strong>  Средне</p>
        <p><strong>Стоимость:</strong>  Платно</p>
        <p><strong>Домен:</strong> ваш-домен.ru</p>
        <p><strong>Плюсы:</strong> Свой домен, больше возможностей</p>
        <button class="action-button"
            onclick="showFileZillaSteps()">Выбрать этот способ</button>
    </div>
</div>
```

💡 Рекомендация для начинающих: Начните с GitHub Pages — это бесплатно, надёжно и даёт Вам важный навык работы с Git.

</div>
</div>

<!-- GitHub Pages инструкция -->

<div class="example-section" id="github-steps" style="display: none;">
 <h2>3. Подробная инструкция: GitHub Pages</h2>

<div class="step-container">

 <div class="step">

 <div class="step-number">1</div>

 <h3>Регистрация на GitHub</h3>

 <p>Что делать: Перейдите на github.com и создайте
аккаунт</p>

 <div class="screenshot">

// Адрес для регистрации:

https://github.com/signup

// Заполните:

• Username (логин) - запомните его!

• Email address

• Password

 </div>

 <button class="action-button"

 onclick="window.open('https://github.com/signup', '_blank')">Перейти к
регистрации</button>

 </div>

<div class="step">

 <div class="step-number">2</div>

 <h3>Создание нового репозитория</h3>

 <p>Репозиторий — это как папка-хранилище
для ваших файлов</p>

 <div class="screenshot">

// После входа в аккаунт:

1. Нажмите "+" в правом верхнем углу

2. Выберите "New repository"

3. Введите имя: ВАШ-ЛОГИН.github.io

4. Поставьте галочку "Add a README file"

5. Нажмите "Create repository"

 </div>

 <div class="warning">

Важно: Имя репозитория ДОЛЖНО быть в формате ваш-логин.github.io

</div>

</div>

<div class="step">

<div class="step-number">3</div>

<h3>Загрузка файлов через веб-интерфейс</h3>

<p>Самый простой способ — перетащить файлы прямо в браузере</p>

<div class="Screenshot">

// В Вашем репозитории:

1. Нажмите "Add file" → "Upload files"
2. Перетащите ВСЕ файлы Вашего сайта в окно
3. ИЛИ нажмите "choose your files" и выберите файлы
4. Прокрутите вниз и нажмите "Commit changes"

</div>

<div class="file-structure">

📁 Что загружать:

Перетащите ВСЕ файлы из Вашей папки:

- index.html

- style.css

- about.html

- ВСЮ папку images/

- Все остальные файлы

</div>

</div>

<div class="step">

<div class="step-number">4</div>

<h3>Включение GitHub Pages</h3>

<p>Активируем хостинг для ваших файлов</p>

<div class="Screenshot">

// В репозитории:

1. Нажмите вкладку "Settings"
2. В меню слева выберите "Pages"
3. В разделе "Source" выберите "Deploy from a branch"
4. Выберите ветку "main"
5. Нажмите "Save"

</div>

</div>

<div class="step">

```
<div class="step-number">5</div>
<h3>🎉 Ваш сайт работает!</h3>
<p><strong>Через 1-2 минуты</strong> Ваш сайт будет доступен по адресу:</p>
```

```
<div class="github-preview" id="github-url">
    https://ваши-логин.github.io
</div>
<p><strong>Проверьте:</strong> Откройте этот адрес в браузере</p>
```

```
<div class="success">
    <strong>✓ Поздравляем!</strong> Ваш сайт теперь в интернете и доступен по постоянному адресу!
</div>
</div>
</div>
</div>
```

```
<!-- Netlify инструкция -->
<div class="example-section" id="netlify-steps" style="display: none;">
    <h2>3. Подробная инструкция: Netlify Drop</h2>
```

```
<div class="step-container">
    <div class="step">
        <div class="step-number">1</div>
        <h3>Переход на Netlify Drop</h3>
        <p><strong>Что делать:</strong> Откройте сайт Netlify Drop</p>
        <div class="screenshot">
```

// Адрес:

<https://app.netlify.com/drop>

```
</div>
    <button class="action-button"
        onclick="window.open('https://app.netlify.com/drop', '_blank')">Открыть Netlify Drop</button>
</div>
```

```
<div class="step">
    <div class="step-number">2</div>
    <h3>Перетаскивание файлов</h3>
    <p><strong>Просто перетащите</strong> всю папку с сайтом в окно браузера</p>
    <div class="screenshot">
```

// Действия:

1. Откройте папку с Вашим сайтом на компьютере
2. Выделите ВСЕ файлы (index.html, style.css, images/ и т.д.)

3. Перетащите их в окно Netlify Drop

4. Отпустите кнопку мыши

```
</div>  
</div>
```

```
<div class="step">  
  <div class="step-number">3</div>  
  <h3>⌚ Ожидание загрузки</h3>  
  <p><strong>Netlify автоматически</strong> обработает Ваши  
файлы</p>  
  <div class="screenshot">
```

// Процесс:

- Файлы загружаются...
- Netlify настраивает сайт...
- Сайт публикуется...
- Готово!

```
</div>  
</div>
```

```
<div class="step">  
  <div class="step-number">4</div>  
  <h3>🎉 Сайт готов!</h3>  
  <p><strong>Netlify выдаст вам</strong> случайный адрес  
вида:</p>  
  <div class="github-preview">  
    https://случайный-набор-букв.netlify.app  
  </div>  
  <p><strong>Скопируйте этот адрес</strong> — это Ваш сайт!  
</p>
```

```
<div class="success">  
  <strong>✅ Всё готово!</strong> Самый простой способ  
разместить сайт в интернете!  
  </div>  
</div>  
</div>  
</div>
```

<!-- FileZilla инструкция -->

```
<div class="example-section" id="filezilla-steps" style="display: none;">  
  <h2>3. Подробная инструкция: FileZilla + платный хостинг</h2>
```

```
<div class="step-container">  
  <div class="step">  
    <div class="step-number">1</div>
```

```
<h3>Подготовка данных от хостинга</h3>
<p><strong>Вам понадобятся данные</strong> от Вашего
хостинг-провайдера:</p>
<div class="Screenshot">
// Данные FTP (обычно в панели управления хостингом):
• Хост (Host): ftp.ваш-сайт.ru или IP-адрес
• Имя пользователя (Username): Ваш логин
• Пароль (Password): Ваш пароль
• Порт (Port): обычно 21
</div>
</div>

<div class="step">
<div class="step-number">2</div>
<h3>Установка FileZilla</h3>
<p><strong>Скачайте и установите</strong> бесплатную
программу FileZilla</p>
<div class="Screenshot">
// Скачать можно здесь:
https://filezilla-project.org

// Установка:
1. Скачайте FileZilla Client
2. Запустите установщик
3. Следуйте инструкциям установки
</div>
<button class="action-button" onclick="window.open('https://filezilla-
project.org', '_blank')">Скачать FileZilla</button>
</div>

<div class="step">
<div class="step-number">3</div>
<h3>Подключение к хостингу</h3>
<p><strong>Введите данные</strong> для подключения</p>
<div class="Screenshot">
// В FileZilla:
1. Введите Хост, Имя пользователя, Пароль, Порт
2. Нажмите "Быстрое соединение"
3. Если данные верны — подключитесь к серверу
</div>
</div>

<div class="step">
<div class="step-number">4</div>
<h3>Загрузка файлов</h3>
```

```
<p><strong>Перетащите файлы</strong> с компьютера на
сервер</p>
<div class="screenshot">
// Процесс загрузки:
1. Слева — файлы на Вашем компьютере
2. Справа — файлы на хостинге
3. Найдите папку public_html или www на сервере
4. Перетащите ВСЕ файлы сайта в эту папку
</div>
</div>

<div class="step">
<div class="step-number">5</div>
<h3>⚠ Проверка сайта</h3>
<p><strong>Откройте Ваш домен</strong> в браузере</p>
<div class="github-preview">
    https://ваш-домен.ru
</div>
<div class="success">
    <strong>✅ Готово!</strong> Ваш сайт на платном хостинге

```

работает!

```
</div>
</div>
</div>
</div>

<!-- Проверка и устранение проблем -->
<div class="example-section">
<h2>4. Проверка работы и устранение проблем</h2>

<div class="visual-guide">
<div class="visual-card">
    <div class="visual-icon">🔍</div>
    <h3>Проверка сайта</h3>
    <p>Откройте Ваш адрес в браузере и проверьте:</p>
    <ul style="text-align: left;">
        <li>✅ Открывается ли главная страница</li>
        <li>✅ Работают ли переходы между страницами</li>
        <li>✅ Отображаются ли картинки</li>
        <li>✅ Применяются ли стили CSS</li>
    </ul>
</div>

<div class="visual-card">
    <div class="visual-icon">⚠</div>
```

```

<h3>Частые проблемы</h3>
<p>Если что-то не работает:</p>
<ul style="text-align: left;">
    <li>✗ Главная страница не index.html</li>
    <li>✗ Неправильные пути к файлам</li>
    <li>✗ Файлы загружены не в корень</li>
    <li>✗ Не прошло время настройки</li>
</ul>
</div>

<div class="visual-card">
    <div class="visual-icon">🔧</div>
    <h3>Решение проблем</h3>
    <p>Что делать если не работает:</p>
    <ul style="text-align: left;">
        <li>✓ Подождите 5-10 минут</li>
        <li>✓ Проверьте имя index.html</li>
        <li>✓ Перезагрузите страницу (Ctrl+F5)</li>
        <li>✓ Перезагрузите файлы</li>
    </ul>
</div>
</div>

<div class="interactive-demo">
    <h3>🎯 Тест-чеклист после загрузки:</h3>
    <textarea placeholder="Отметьте что проверили:">
        [ ] Главная страница открывается
        [ ] Все страницы доступны
        [ ] Картинки отображаются
        [ ] Стили CSS работают
        [ ] Ссылки между страницами работают
        [ ] Сайт быстро загружается" style="width: 100%; height: 150px; padding: 10px; border: 1px solid #ddd; border-radius: 5px; margin: 10px 0;"></textarea>
    </div>
</div>

<!-- Что делать после загрузки -->
<div class="example-section">
    <h2>5. Что делать после успешной загрузки?</h2>

    <div style="display: grid; grid-template-columns: repeat(auto-fit, minmax(250px, 1fr)); gap: 15px; margin: 20px 0;">
        <div style="border: 2px solid #3498db; padding: 15px; border-radius: 8px; text-align: center;">
```

```
<div style="font-size: 2em;">📱 </div>
<h4>Проверить на телефоне</h4>
<p>Откройте сайт на смартфоне — всё ли хорошо
отображается?</p>
</div>

<div style="border: 2px solid #27ae60; padding: 15px; border-radius:
8px; text-align: center;">
    <div style="font-size: 2em;">👤 </div>
    <h4>Показать друзьям</h4>
    <p>Отправьте ссылку друзьям и попросите проверить</p>
</div>

<div style="border: 2px solid #e74c3c; padding: 15px; border-radius:
8px; text-align: center;">
    <div style="font-size: 2em;">📝 </div>
    <h4>Сохранить адрес</h4>
    <p>Запишите адрес сайта — он Вам ещё пригодится!</p>
</div>

<div style="border: 2px solid #f39c12; padding: 15px; border-radius:
8px; text-align: center;">
    <div style="font-size: 2em;">🎉 </div>
    <h4>Поздравить себя</h4>
    <p>Вы только что разместили свой сайт в интернете!</p>
</div>
</div>

<div class="success">
    <strong>🏆 Поздравляем!</strong> Вы только что прошли один из
самых важных этапов в веб-разработке — разместили свой сайт в
интернете. Теперь весь мир может увидеть Вашу работу!
    </div>
</div>

<script>
    // Функции для переключения между способами загрузки
    function showGitHubSteps() {
        document.getElementById('github-steps').style.display = 'block';
        document.getElementById('netlify-steps').style.display = 'none';
        document.getElementById('filezilla-steps').style.display = 'none';
    }

    function showNetlifySteps() {
        document.getElementById('github-steps').style.display = 'none';

```

```
document.getElementById('netlify-steps').style.display = 'block';
document.getElementById('filezilla-steps').style.display = 'none';
}

function showFileZillaSteps() {
    document.getElementById('github-steps').style.display = 'none';
    document.getElementById('netlify-steps').style.display = 'none';
    document.getElementById('filezilla-steps').style.display = 'block';
}

// Обновление URL для GitHub
const githubUrl = document.getElementById('github-url');
// В реальном приложении здесь можно было бы подставить
реальный логин пользователя
</script>
</body>
</html>
```

Часть 7: Что Вы теперь умеете

- Подготавливать файлы сайта к загрузке
- Выбирать подходящий способ загрузки для ваших задач
- Использовать GitHub Pages для бесплатного размещения
- Работать с Netlify Drop для сверх-простой загрузки
- Настраивать FileZilla для работы с платным хостингом
- Проверять работоспособность сайта после загрузки
- Устранять типичные проблемы при размещении сайта

Часть 8: Итоги и рекомендации

Ключевые моменты успешной загрузки:

Главная страница ДОЛЖНА называться **index.html** — это стандарт
Все файлы должны быть в одной папке — сохраняйте структуру
Проверяйте сайт локально перед загрузкой
Используйте правильные пути к картинкам и CSS файлам
Наберитесь терпения — некоторым хостингам нужно время на настройку

Профессиональные советы:

Начните с GitHub Pages — это бесплатно и даёт ценный опыт
Сохраняйте копии файлов на компьютере — всегда имейте резервную копию
Тестируйте на разных устройствах — проверяйте как выглядит на телефоне

**Делитесь ссылкой с друзьями — получайте обратную связь
Не бойтесь ошибок — всегда можно перезагрузить файлы заново**

 **Что делать после успешной загрузки:**

**Добавьте сайт в портфолио — это Ваша первая работа в интернете!
Продолжайте улучшать — теперь Вы можете обновлять сайт когда
захотите
Экспериментируйте — пробуйте разные дизайны и функции
Изучайте дальше — JavaScript, анимации, адаптивную вёрстку
Создавайте новые проекты — примените знания на практике**

**Помните: Первая загрузка сайта в интернет — это как первый шаг
ребёнка. Он может быть неуверенным, но это начало большого
пути! **

**Теперь Ваш сайт живёт в интернете и доступен из любой точки
мира — это огромное достижение! **

13.4. Ваш сайт в сети! Дайте ссылку детям и друзьям

Введение: Ваш голос услышен в цифровом мире!

Представьте, что Вы только что опубликовали свою первую книгу — теперь пришло время рассказать о ней всем! Ваш сайт в интернете — это Ваша первая "публикация", и сегодня мы научимся правильно ею делиться с близкими и получать ценные отзывы.

Часть 1: Ваш сайт живёт в интернете — осознаём значимость

1.1. Что только что произошло?

html

```
<div class="achievement">
  <h3>🎉 Вы только что совершили важный шаг!</h3>
  <div class="milestone">
    <strong>Раньше:</strong> Файлы на Вашем компьютере<br>
    <strong>Теперь:</strong> Полноценный сайт в интернете<br>
    <strong>Результат:</strong> Доступен из любой точки мира!
  </div>
</div>
```

1.2. Где теперь живёт Ваш сайт:

text

🌐 Ваш САЙТ ТЕПЕРЬ ДОСТУПЕН ПО АДРЕСУ:

```
https://ваш-логин.github.io
ИЛИ
https://случайный-адрес.netlify.app
ИЛИ
https://ваш-красивый-домен.ru
```

Часть 2: Как правильно поделиться ссылкой — 5 лучших способов

2.1. Способ 1: Простая отправка в мессенджерах

html

```
<div class="sharing-method">
  <h4>📱 В мессенджерах (WhatsApp, Telegram)</h4>
  <div class="message-template">
    <strong>Пример сообщения:</strong><br>
    "Привет! Я создал(а) свой первый сайт 🎉<br>
    Посмотри, пожалуйста: [ваша ссылка]<br>
```

Напиши, что думаешь! 😊"
</div>
</div>

2.2. Способ 2: Email рассылка для родственников

html

```
<div class="sharing-method">
  <h4>✉ Email для старшего поколения</h4>
  <div class="email-template">
    <strong>Тема:</strong> Мой первый сайт!<br>
    <strong>Текст:</strong><br>
    "Дорогие мама и папа!<br>
    Я научился(ась) создавать сайты и сделал(а) свой первый
    проект.<br>
    Вот ссылка: [ваша ссылка]<br>
    Буду рад(а) Вашим отзывам!"<br>
  </div>
</div>
```

Часть 3: Полная страница с инструкциями по разделению

html

```
<!DOCTYPE html>
<html lang="ru">
<head>
  <meta charset="utf-8">
  <title>Ваш сайт в сети! Делимся с миром</title>
  <style>
    body {
      font-family: 'Arial', sans-serif;
      max-width: 1200px;
      margin: 0 auto;
      padding: 20px;
      line-height: 1.6;
      background: linear-gradient(135deg, #667eea 0%, #764ba2 100%);
      min-height: 100vh;
    }
    .container {
      background: white;
      border-radius: 15px;
      box-shadow: 0 10px 30px rgba(0,0,0,0.2);
      overflow: hidden;
    }
    .header-section {
```

```
background: linear-gradient(135deg, #2c3e50, #3498db);
color: white;
padding: 40px 30px;
text-align: center;
}
.content-section {
  padding: 30px;
}
.example-section {
  background: white;
  padding: 25px;
  margin: 25px 0;
  border-radius: 10px;
  box-shadow: 0 2px 5px rgba(0,0,0,0.1);
  border: 2px solid #e0e0e0;
}
h1, h2 {
  color: #2c3e50;
}
.header-section h1 {
  color: white;
  font-size: 2.5em;
  margin-bottom: 10px;
}
.code-example {
  background: #2d2d2d;
  color: #f8f8f2;
  padding: 15px;
  border-radius: 5px;
  font-family: 'Courier New', monospace;
  margin: 15px 0;
  overflow-x: auto;
}
.note {
  background: #e7f3ff;
  padding: 15px;
  border-left: 4px solid #2196F3;
  margin: 15px 0;
  border-radius: 5px;
}
.warning {
  background: #fff3e0;
  padding: 15px;
  border-left: 4px solid #ff9800;
  margin: 15px 0;
```

```
    border-radius: 5px;
}
.success {
    background: #e8f5e8;
    padding: 15px;
    border-left: 4px solid #4CAF50;
    margin: 15px 0;
    border-radius: 5px;
}
.celebration {
    background: linear-gradient(135deg, #ff9a9e, #fad0c4);
    padding: 20px;
    border-radius: 10px;
    text-align: center;
    margin: 20px 0;
}
.sharing-methods {
    display: grid;
    grid-template-columns: repeat(auto-fit, minmax(300px, 1fr));
    gap: 20px;
    margin: 20px 0;
}
.method-card {
    background: white;
    border: 2px solid #3498db;
    border-radius: 10px;
    padding: 20px;
    transition: transform 0.3s ease;
}
.method-card:hover {
    transform: translateY(-5px);
}
.method-icon {
    font-size: 2.5em;
    margin-bottom: 15px;
    text-align: center;
}
.url-display {
    background: #f8f9fa;
    border: 2px dashed #3498db;
    border-radius: 8px;
    padding: 15px;
    margin: 15px 0;
    font-family: 'Courier New', monospace;
    font-size: 1.2em;
```

```
    text-align: center;
    word-break: break-all;
}
.message-templates {
    background: #f5f5f5;
    padding: 20px;
    border-radius: 8px;
    margin: 20px 0;
}
.template {
    background: white;
    padding: 15px;
    margin: 10px 0;
    border-radius: 5px;
    border-left: 4px solid #3498db;
}
.interactive-demo {
    background: #e3f2fd;
    padding: 20px;
    border-radius: 8px;
    margin: 20px 0;
}
.action-button {
    background: #3498db;
    color: white;
    padding: 12px 25px;
    border: none;
    border-radius: 5px;
    cursor: pointer;
    font-size: 16px;
    margin: 10px 5px;
    transition: background 0.3s ease;
}
.action-button:hover {
    background: #2980b9;
}
.button-success {
    background: #27ae60;
}
.button-success:hover {
    background: #219653;
}
.testimonial {
    background: #fff3e0;
    padding: 20px;
```

```
border-radius: 8px;
margin: 20px 0;
border-left: 4px solid #f39c12;
}
.qr-section {
text-align: center;
padding: 20px;
}
.qr-placeholder {
width: 200px;
height: 200px;
background: #f0f0f0;
margin: 0 auto;
border: 2px dashed #ccc;
display: flex;
align-items: center;
justify-content: center;
color: #666;
}
.feedback-guide {
display: grid;
grid-template-columns: repeat(auto-fit, minmax(250px, 1fr));
gap: 15px;
margin: 20px 0;
}
.feedback-card {
background: white;
border: 1px solid #e0e0e0;
border-radius: 8px;
padding: 15px;
}

```

</style>

```
</head>
<body>
<div class="container">
<!-- Шапка с поздравлением --&gt;
&lt;div class="header-section"&gt;
&lt;h1&gt;<img alt="cake icon" data-bbox="188 778 258 800"/> Ваш сайт в сети!</h1>
<p style="font-size: 1.3em; opacity: 0.9;">Поздравляем! Теперь
поделитесь своим творением с миром</p>
</div>

<div class="content-section">
<!-- Церемония поздравления --&gt;
&lt;div class="celebration"&gt;</pre>
```

```
<h2 style="color: #d63031; margin-top: 0;">🏆 Вы это сделали!
</h2>
<p style="font-size: 1.2em; font-weight: bold;">От файлов на
компьютере до полноценного сайта в интернете — это огромное
достижение!</p>
</div>

<!-- Отображение Вашего URL -->
<div class="example-section">
<h2>1. Ваш сайт теперь живёт здесь:</h2>

<div class="url-display" id="website-url">
    https://ваш-логин.github.io
</div>

<div class="note">
    <strong>💡 Скопируйте эту ссылку</strong> — это адрес
Вашего сайта в интернете. Теперь любой человек в мире может увидеть
Вашу работу, введя этот адрес в браузере.
</div>

<div style="text-align: center; margin: 20px 0;">
    <button class="action-button" onclick="copyUrl()">📋
Скопировать ссылку</button>
    <button class="action-button button-success"
onclick="testWebsite()">🔍 Проверить сайт</button>
</div>
</div>

<!-- Способы поделиться -->
<div class="example-section">
<h2>2. 5 способов поделиться с друзьями и семьёй</h2>

<div class="sharing-methods">
    <div class="method-card">
        <div class="method-icon">📱</div>
        <h3>Мессенджеры</h3>
        <p><strong>WhatsApp, Telegram, Viber</strong></p>
        <p>Быстрая отправка близким друзьям и родным</p>
        <button class="action-button"
onclick="showMessengerTemplate()">Показать шаблон</button>
    </div>
    <div class="method-card">
        <div class="method-icon">✉️</div>
```

```
<h3>Email</h3>
<p><strong>Для старших родственников</strong></p>
<p>Подробное письмо с объяснениями</p>
<button class="action-button"
onclick="showEmailTemplate()">Показать шаблон</button>
</div>

<div class="method-card">
<div class="method-icon"> </div>
<h3>Соцсети</h3>
<p><strong>ВКонтакте, Одноклассники</strong></p>
<p>Поделиться с широким кругом знакомых</p>
<button class="action-button"
onclick="showSocialTemplate()">Показать шаблон</button>
</div>

<div class="method-card">
<div class="method-icon"> </div>
<h3>По телефону</h3>
<p><strong>Звонок друзьям</strong></p>
<p>Личное общение + отправка ссылки</p>
<button class="action-button"
onclick="showCallTemplate()">Показать шаблон</button>
</div>

<div class="method-card">
<div class="method-icon"> </div>
<h3>QR-код</h3>
<p><strong>Для личной встречи</strong></p>
<p>Показать код на телефоне — перейти на сайт</p>
<button class="action-button"
onclick="showQRCode()">Сгенерировать QR-код</button>
</div>
</div>

<!-- Шаблоны сообщений -->
<div class="example-section" id="templates-section" style="display: none;">
<h2>3. Готовые шаблоны сообщений</h2>

<div class="message-templates">
<div class="template" id="messenger-template" style="display: none;">
<h4> Для мессенджеров:</h4>
```

```
<div class="code-example">
```

Привет! 😊

Я только что создал(а) свой первый сайт! 🎉

Это мой учебный проект по HTML.

Посмотри, пожалуйста: [ВАША ССЫЛКА]

Напиши, что думаешь! Буду рад(а) обратной связи.

P.S. Если что-то не работает — тоже напиши 🙏

```
</div>
```

```
<button class="action-button"
```

```
onclick="copyTemplate('messenger')">&copy; Скопировать текст</button>
```

```
<div class="template" id="email-template" style="display: none;">
```

```
<h4>✉️ Для email (родителям, родственникам):</h4>
```

```
<div class="code-example">
```

Дорогие мама и папа!

Хочу поделиться с Вами своей радостью — я научился(ась) создавать сайты и сделал(а) свой первый проект!

Это сайт о [тема Вашего сайта], который я создал(а) с нуля. Я изучал(а) HTML и CSS, и это мой первый результат.

Вот ссылка на сайт: [ВАША ССЫЛКА]

Буду очень рад(а), если посмотрите и напишете ваше мнение!

Для меня это очень важно.

С любовью,

[Ваше имя]

```
</div>
```

```
<button class="action-button"
```

```
onclick="copyTemplate('email')">&copy; Скопировать текст</button>
```

```
<div class="template" id="social-template" style="display: none;">
```

```
<h4>👤 Для социальных сетей:</h4>
```

```
<div class="code-example">
```

Друзья! Изучаю веб-разработку и создал(а) свой первый сайт! 🚀

Это [описание сайта] — мой учебный проект по HTML/CSS.

Ссылка: [ВАША ССЫЛКА]

Буду благодарен(на) за обратную связь и советы!

Особенно интересует:

- Как отображается на телефоне?
- Всё ли понятно на сайте?
- Что можно улучшить?

#вебразработка #первыйсайт #html #css

</div>

<button class="action-button"

onclick="copyTemplate('social')">© Скопировать текст</button>

</div>

<div class="template" id="call-template" style="display: none;">

<h4>☎ Для телефонного разговора:</h4>

<div class="code-example">

Что сказать:

"Привет! Я хочу показать тебе кое-что крутое —
я создал(а) свой первый сайт! Отправляю тебе ссылку,
посмотри, когда будет время. Мне важно твоё мнение!"

После отправки:

"Получил(а) ссылку? Зайди, пожалуйста, и скажи,
как тебе дизайн и всё ли работает нормально."

</div>

<button class="action-button"

onclick="copyTemplate('call')">© Скопировать скрипт</button>

</div>

</div>

</div>

<!-- QR-код секция -->

<div class="example-section" id="qr-section" style="display: none;">

<h2>4. QR-код для Вашего сайта</h2>

<div class="qr-section">

<div class="qr-placeholder" id="qr-code">

QR-код появится здесь

</div>

<p style="margin-top: 15px;">Покажите этот код друзьям —

они смогут отсканировать его камерой телефона и перейти на Ваш сайт!
</p>

<button class="action-button" onclick="generateQRCode()">©</button>

Сгенерировать QR-код</button>

```
</div>
</div>

<!-- Как получать обратную связь -->
<div class="example-section">
  <h2>5. Как получать полезные отзывы</h2>

  <div class="warning">
    <strong>🎯 Важно:</strong> Просто "мне нравится" — это
    приятно, но бесполезно для развития.
    Попросите конкретную обратную связь!
  </div>

  <div class="feedback-guide">
    <div class="feedback-card">
      <h4>❓ Какие вопросы задавать</h4>
      <ul>
        <li>"Что сразу бросилось в глаза?"</li>
        <li>"Всё ли понятно на сайте?"</li>
        <li>"Удобно ли читать текст?"</li>
        <li>"Что бы ты улучшил(а)?"</li>
      </ul>
    </div>

    <div class="feedback-card">
      <h4>📱 Проверка на устройствах</h4>
      <ul>
        <li>Попросите проверить на телефоне</li>
        <li>На планшете</li>
        <li>На компьютере</li>
        <li>В разных браузерах</li>
      </ul>
    </div>

    <div class="feedback-card">
      <h4>🎨 Дизайн и удобство</h4>
      <ul>
        <li>Нравится ли цветовая гамма?</li>
        <li>Удобно ли расположены элементы?</li>
        <li>Не "режет" ли глаза что-то?</li>
        <li>Логично ли расположен контент?</li>
      </ul>
    </div>
  </div>
</div>
```

```
<div class="interactive-demo">
  <h4>📝 Чеклист для обратной связи:</h4>
  <textarea id="feedback-checklist" style="width: 100%; height: 120px; padding: 10px; border: 1px solid #ddd; border-radius: 5px;" placeholder="Записывайте сюда отзывы:">
    • От мамы: ...
    • От друга: ...
    • От коллеги: ...
    • Что нужно исправить: ...</textarea>
```

```
    <button class="action-button" onclick="saveFeedback()">💾 Сохранить отзывы</button>
```

```
<!-- Что делать с отзывами -->
```

```
<div class="example-section">
  <h2>6. Что делать с полученными отзывами?</h2>
```

```
<div style="display: grid; grid-template-columns: repeat(auto-fit, minmax(250px, 1fr)); gap: 15px; margin: 20px 0;">
```

```
  <div style="border: 2px solid #3498db; padding: 15px; border-radius: 8px; text-align: center;">
```

```
    <div style="font-size: 2em;">👍</div>
```

```
    <h4>Положительные отзывы</h4>
```

```
    <p>Сохраните их для мотивации! Это доказательство, что
```

```
Вы движетесь в правильном направлении.</p>
```

```
</div>
```

```
<div style="border: 2px solid #f39c12; padding: 15px; border-radius: 8px; text-align: center;">
```

```
    <div style="font-size: 2em;">💡</div>
```

```
    <h4>Конструктивная критика</h4>
```

```
    <p>Это золото! Составьте список улучшений и постепенно их реализуйте.</p>
```

```
</div>
```

```
<div style="border: 2px solid #e74c3c; padding: 15px; border-radius: 8px; text-align: center;">
```

```
    <div style="font-size: 2em;">🐞</div>
```

```
    <h4>Найденные ошибки</h4>
```

```
    <p>Исправляйте их в первую очередь. Каждая исправленная ошибка — шаг к профессионализму.</p>
```

```
</div>
```

```
<div style="border: 2px solid #27ae60; padding: 15px; border-radius: 8px; text-align: center;">
    <div style="font-size: 2em;"> </div>
    <h4>Идеи для улучшений</h4>
    <p>Записывайте все идеи. Они станут основой для ваших следующих проектов.</p>
</div>
</div>
</div>

<!-- Следующие шаги -->
<div class="example-section">
    <h2>7. Что дальше? Ваши следующие шаги</h2>

    <div class="success">
        <h3> Вы прошли огромный путь!</h3>
        <p>От первого тега HTML до работающего сайта в интернете — это настоящее достижение!</p>
    </div>

    <div class="note">
        <strong> План развития:</strong>
        <ol>
            <li><strong>Соберите отзывы</strong> от 5-10 человек</li>
            <li><strong>Исправьте найденные ошибки</strong></li>
            <li><strong>Добавьте улучшения</strong> на основе обратной связи</li>
            <li><strong>Создайте новый проект</strong> с учётом полученного опыта</li>
            <li><strong>Продолжайте учиться</strong> — JavaScript, адаптивная вёрстка, фреймворки</li>
        </ol>
    </div>

    <div class="testimonial">
        <h4> Запомните этот момент!</h4>
        <p>Сегодня Вы разместили свой первый сайт в интернете. Через год Вы будете с улыбкой вспоминать этот день, глядя на свои более сложные и совершенные проекты.</p>
        <p style="text-align: right; font-style: italic; margin-top: 15px;">— Гордитесь собой! Вы это заслужили </p>
    </div>
</div>
</div>
```

```
<script>
    // Основные функции для работы с ссылкой
    function copyUrl() {
        const url = document.getElementById('website-url').textContent;
        navigator.clipboard.writeText(url).then(() => {
            alert('✅ Ссылка скопирована в буфер обмена!');
        });
    }

    function testWebsite() {
        const url = document.getElementById('website-url').textContent;
        window.open(url, '_blank');
    }

    // Функции для шаблонов сообщений
    function showMessengerTemplate() {
        document.getElementById('templates-section').style.display = 'block';
        document.getElementById('messenger-template').style.display =
        'block';
        document.getElementById('email-template').style.display = 'none';
        document.getElementById('social-template').style.display = 'none';
        document.getElementById('call-template').style.display = 'none';
    }

    function showEmailTemplate() {
        document.getElementById('templates-section').style.display = 'block';
        document.getElementById('messenger-template').style.display =
        'none';
        document.getElementById('email-template').style.display = 'block';
        document.getElementById('social-template').style.display = 'none';
        document.getElementById('call-template').style.display = 'none';
    }

    function showSocialTemplate() {
        document.getElementById('templates-section').style.display = 'block';
        document.getElementById('messenger-template').style.display =
        'none';
        document.getElementById('email-template').style.display = 'none';
        document.getElementById('social-template').style.display = 'block';
        document.getElementById('call-template').style.display = 'none';
    }

    function showCallTemplate() {
        document.getElementById('templates-section').style.display = 'block';
    }
```

```
document.getElementById('messenger-template').style.display =
'none';
document.getElementById('email-template').style.display = 'none';
document.getElementById('social-template').style.display = 'none';
document.getElementById('call-template').style.display = 'block';
}

function copyTemplate(type) {
    const template = document.getElementById(`#${type}-template`).querySelector('.code-example').textContent;
    const url = document.getElementById('website-url').textContent;
    const textWithUrl = template.replace('[ВАША ССЫЛКА]', url);

    navigator.clipboard.writeText(textWithUrl).then(() => {
        alert('✅ Текст скопирован! Вставьте его в сообщение.');
    });
}

// Функции для QR-кода
function showQRCode() {
    document.getElementById('qr-section').style.display = 'block';
    generateQRCode();
}

function generateQRCode() {
    const url = document.getElementById('website-url').textContent;
    const qrPlaceholder = document.getElementById('qr-code');

    // В реальном приложении здесь была бы генерация настоящего
    // QR-кода
    // Для демонстрации создаём простой плейсхолдер
    qrPlaceholder.innerHTML =
        `

<div style="font-size: 3em;">📱</div>
            <div style="font-size: 0.8em; margin-top: 10px;">QR-код
для:<br>${url}</div>
        </div>
    `;
}

// Сохранение отзывов
function saveFeedback() {
    const feedback = document.getElementById('feedback-
checklist').value;
    if (feedback.trim()) {


```

```

        alert('✅ Отзывы сохранены! Вернитесь к ним позже для работы над улучшениями.');
        // В реальном приложении здесь можно сохранять в localStorage
    }
    localStorage.setItem('websiteFeedback', feedback);
} else {
    alert('⚠ Сначала запишите отзывы в поле выше');
}
}

// Загрузка сохранённых отзывов при загрузке страницы
window.addEventListener('load', function() {
    const savedFeedback = localStorage.getItem('websiteFeedback');
    if (savedFeedback) {
        document.getElementById('feedback-checklist').value =
    savedFeedback;
    }
}

// В реальном приложении здесь можно подставить реальный URL пользователя
// const userUrl = localStorage.getItem('userWebsiteUrl') ||
'https://ваши-логин.github.io';
// document.getElementById('website-url').textContent = userUrl;
});
</script>
</body>
</html>

```

Часть 7: Что Вы теперь умеете

- Осознавать значимость размещения сайта в интернете**
- Копировать и делиться ссылкой на Ваш сайт**
- Использовать разные способы распространения ссылки**
- Составлять эффективные сообщения для разных аудиторий**
- Генерировать QR-коды для быстрого доступа**
- Собирать конструктивную обратную связь**
- Анализировать и использовать полученные отзывы**

Часть 8: Итоги и рекомендации

 **Ключевые моменты успешного распространения:**
Гордитесь своим достижением — создание работающего сайта это серьёзный успех
Делитесь с близкими в первую очередь — их поддержка важна
Просите конкретную обратную связь — это ускорит ваше развитие

**Сохраняйте все отзывы — они станут Вашим гидом для улучшений
Не бойтесь критики — она делает Вас сильнее**

Психологические аспекты первого сайта:

**Это нормально волноваться — Вы делитесь частью себя
Разные люди увидят разное — кто-то дизайн, кто-то содержание
Критика ≠ личная оценка — это обратная связь о проекте, не о вас
Каждый комментарий ценен — даже "не понравилось" помогает расти
Это только начало — Ваш первый сайт, но не последний**

Практические советы по распространению:

**Начните с 2-3 самых близких людей — они дадут самую мягкую обратную связь
Используйте шаблоны из урока — они уже протестированы и работают
Сохраняйте скриншоты похвал — для мотивации в сложные моменты
Создайте папку "Мои проекты" — сохраняйте там все свои работы
Расскажите о процессе — людям интересно как Вы это сделали**

Что говорят те, кто прошёл этот путь:

"Помню, как отправлял ссылку на свой первый сайт маме. Она позвонила в слезах: 'Сынок, ты в интернете!' Этот момент я никогда не забуду." — Артём, веб-разработчик
"Первые отзывы друзей помогли мне понять, что нужно учиться адаптивной вёрстке. Все смотрели с телефонов, а сайт был только для компьютера!" — Марина, фронтенд-разработчик
"Свой первый сайт я сделала в 45 лет. Когда поделилась с подругами, они не поверили, что это возможно в нашем возрасте. Теперь учу их!" — Ольга, начинающий разработчик

Помните: Сегодня Вы не просто разместили сайт в интернете — Вы заявили о себе в цифровом мире и сделали первый шаг к карьере веб-разработчика! 

Ваш сайт теперь живёт в интернете, а Ваши навыки будут расти с каждым новым проектом. Гордитесь собой — Вы это заслужили! 

Приложения

Приложение А: Шпаргалка по основным тегам HTML

Дорогие друзья! Эта шпаргалка — Ваш быстрый помощник. Здесь собраны самые важные и часто используемые теги. Не нужно запоминать их все сразу! Просто держите эту таблицу под рукой и заглядывайте в нее, когда что-то забыли.

Как пользоваться таблицей:

Тег: Как пишется команда в угловых скобках.

Для чего нужен: Простое и понятное объяснение.

Пример: Как это будет выглядеть в коде.

Как выглядит в браузере: Как это увидит пользователь на странице.

Основные теги (Каркас страницы)

Это основа любой HTML-страницы. Можно сказать, "коробка", в которую всё складывается.

Тег	Для чего нужен	Пример	Как выглядит в браузере
<code><!DOCTYPE html></code>	Объявляет браузеру, что это современный HTML-документ. Всегда должен быть первой строкой!	<code><!DOCTYPE html></code>	(Не виден, это служебная информация)
<code><html></code>	Главный контейнер для всего содержимого страницы.	<code><html> ...вся страница...
<="" code="" html><=""/></code>	(Не виден)
<code><head></code>	"Служебная голова" документа. Здесь хранится заголовок, ссылки на стили и другая техническая информация,	<code><head> <title>Мой сайт</title> </head></code>	(Содержимое не отображается на самой странице, но заголовок <code>title</code> виден в названии вкладки браузера)

	которую не видят посетители.		
<title>	Заголовок окна или вкладки браузера. Очень важен для поисковых систем.	<title>Мой дневник</title>	Виден надписью на вкладке браузера
<body>	Тело документа. Здесь находится всё, что Вы видите на странице: текст, картинки, ссылки и т.д.	<body> <h1>Добро пожаловать!</h1> > <p>Это мой сайт.</p> </body>	Добро пожаловать! Это мой сайт.

Теги для текста и заголовков

С помощью этих тегов Вы структурируете свой текст, делая его удобным для чтения.

Тег	Для чего нужен	Пример	Как выглядит в браузере
<h1>, <h2>, ... <h6>	Заголовки разного уровня. <h1> — самый главный (крупный), <h6> — самый мелкий.	<h1>Глава 1</h1> <h2>Раздел 1.1</h2> <h3>1.1.1 Подраздел</h3>	<h1>Глава 1</h1> <h2>Раздел 1.1</h2> <h3>1.1.1 Подраздел</h3>
<p>	Абзац текста. Браузер автоматически добавляет отступы сверху и снизу.	<p>Это первый абзац.</p> <p>А это второй.</p>	Это первый абзац. А это второй.
 	Перенос строки. Тег одиночный (не требует закрывающего). Просто переходит на новую строку.	Строчка первая. Строчка вторая.	Строчка первая. Строчка вторая.

 или 	Жирный текст. также означает "важный" с точки зрения смысла.	Это жирный текст.	Это жирный текст.
<i> или 	Курсивный текст. также означает "акцентированный" с точки зрения смысла.	Это <i>курсивный</i> текст.	Это курсивный текст.
<mark>	Выделяет текст цветным маркером (обычно желтым).	Это <mark>важная часть</mark> текста.	Это важная часть текста.

Ссылки и изображения

Без этих тегов не было бы Всемирной паутины!

Тег	Для чего нужен	Пример	Как выглядит в браузере
<a>	Создает ссылку (гиперссылку) на другую страницу или сайт. Обязательно используйте атрибут href="адрес" .	Читайте на любимом сайте.	Читайте на любимом сайте.
	Вставляет изображение на страницу. Это одиночный тег. Обязательные атрибуты: src="путь_к_картинке" и alt="описание" .		![Изображение кота]

Списки

Помогают красиво оформить перечисления.

Тег	Для чего нужен	Пример	Как выглядит в браузере
	Создает маркированный список (с точками или другими значками).	 Яблоки Груши 	• Яблоки • Груши
	Создает нумерованный список (с цифрами или буквами).	 Вскипятить воду Заварить чай 	1. Вскипятить воду 2. Заварить чай
	Элемент списка. Используется вн утри или .	(Смотри примеры выше)	(Смотри примеры выше)

Структурные блоки (Макет страницы)

Эти теги не меняют вид текста, но помогают логически разделить страницу на блоки (шапка, меню, основной контент, подвал). Это очень важно для последующего оформления с помощью CSS.

Тег	Для чего нужен	Пример
<div>	Универсальный блочн ый контейнер . Используется для группировки элементов и создания макета.	<div> <h2>Новости</h2> <p>Текст новости...</p> </div>
<header>	"Шапка" сайта или раздела. Обычно здесь находится логотип и главное меню.	<header> <h1>Мой Блог</h1> </header>
<nav>	Раздел для навигационных ссылок (главное меню сайта).	<nav> Главная Обо мне </nav>

<code><main></code>	Обозначает основное, уникальное содержимое страницы. На странице должен быть только один такой тег.	<code><main></code> <code><h2>Статья о садоводстве</h2></code> <code><p>Текст статьи...</p></code> <code></main></code>
<code><footer></code>	"Подвал" сайта или раздела. Часто содержит контактную информацию, авторские права.	<code><footer></code> <code><p>© 2024, Мой сайт</p></code> <code></footer></code>

Небольшой совет

Не стремитесь запомнить всё и сразу. Начните с самого главного: `<html>`, `<head>`, `<body>`, `<h1>`, `<p>`, `<a>` и ``. Этого уже достаточно, чтобы создать свою первую простую страничку! Постепенно, по мере практики, Вы будете запоминать и другие теги. Удачи в освоении веб-грамотности

Приложение Б: Словарик терминов

Дорогие друзья! Осваивая что-то новое, мы часто сталкиваемся с незнакомыми словами. Этот словарик создан специально для того, чтобы Вы могли быстро найти и понять любой термин, связанный с созданием сайтов. Не пугайтесь этих слов — за каждым из них стоит простое и логичное понятие.

A

Атрибут

Простое объяснение: Это **дополнительная инструкция** или **характеристика** внутри тега. Если тег — это слово команды, то атрибут — это уточнение, как именно выполнить эту команду.

Жизненная аналогия: Представьте, что Вы даете команду "варить!" (это тег). Атрибуты — это уточнения: "варить 10 минут (атрибут `time="10"`) и в кастрюле (атрибут `pot="yes"`)".

Пример в HTML: В теге `` слова `src` и `alt` — это атрибуты. Они сообщают браузеру, где найти картинку и что это за картинка.

Браузер

Простое объяснение: Это **программа-переводчик**, которая умеет читать HTML-код и превращать его в красивую, удобную для чтения страницу с текстом, картинками и ссылками.

Жизненная аналогия: Браузер — это как телевизор. Вы не смотрите на внутренние микросхемы и сигналы (HTML-код), Вы смотрите на готовую, понятную картинку (веб-страницу). Примеры браузеров: Google Chrome, Яндекс.Браузер, Mozilla Firefox, Safari.

Важно: Именно браузер показывает Вам сайты в интернете.

B

Веб-страница

Простое объяснение: Один отдельный документ в интернете. Например, статья в онлайн-газете или Ваша анкета в социальной сети.
Жизненная аналогия: Одна страница в книге.

Веб-сайт (или просто "Сайт")

Простое объяснение: Это несколько **веб-страниц**, объединенных вместе под одним общим названием (адресом) и общей темой.

Жизненная аналогия: Целая книга, состоящая из многих страниц (веб-страниц). Например, весь интернет-магазин "Озон" — это сайт, а страница с конкретным чайником — это веб-страница.

Верстка

Простое объяснение: Процесс создания веб-страницы с помощью HTML и CSS. Верстальщик, как конструктор, расставляет на странице текст, картинки, кнопки и другие элементы.

Жизненная аналогия: Процесс макетирования газеты или журнала, когда редактор расставляет статьи, фотографии и заголовки по полосам.

Д

Домен (Доменное имя)

Простое объяснение: Это красивое и удобное для запоминания название сайта, которое Вы вводите в адресную строку браузера.

Жизненная аналогия: Доменное имя — это название улицы и номер дома в интернете. Настоящий "адрес" сайта — это длинный набор цифр (IP-адрес), но людям удобнее использовать имя, например, yandex.ru, а не 77.88.55.60.

Пример: google.com, pensioner.ru, moj-sad.ru.

И

Интернет

Простое объяснение: Всемирная сеть, которая соединяет миллиарды компьютеров по всему миру. Это как гигантская паутина из проводов и беспроводных сигналов.

Жизненная аналогия: Если сайт — это магазин, то интернет — это весь город с его улицами, по которым Вы до этого магазина добираетесь.

К

Код

Простое объяснение: Набор специальных команд и инструкций, написанных на понятном для компьютера языке. HTML — это один из таких языков кода.

Жизненная аналогия: Кулинарный рецепт. В рецепте тоже есть свои "команды" ("взбить", "посолить", "выпекать 30 минут"), следуя которым, Вы получаете готовое блюдо (веб-страницу).

Контейнер

Простое объяснение: Тег (например, <div>), который служит "коробкой" для группировки других тегов. Он помогает организовать элементы на странице.

Жизненная аналогия: Ящик на кухне, в котором Вы храните все столовые приборы. Вместо того чтобы разбрасывать ложки и вилки по всему столу, Вы кладете их в один ящик-контейнер.

С

Сервер

Простое объяснение: Это очень мощный и постоянно работающий компьютер, который хранит файлы веб-сайтов и отдает их Вашим браузерам по запросу.

Жизненная аналогия: В библиотеке много книг (сайтов), но Вы не ходите по книжным полкам сами. Библиотекарь (сервер) по Вашей просьбе находит нужную книгу и приносит её Вам (в браузер).

Ссылка (Гиперссылка)

Простое объяснение: Особый элемент на веб-странице (обычно выделенный синим цветом и подчеркиванием), при щелчке на который Вы переходите на другую страницу или сайт.

Жизненная аналогия: Сноска или цитата в книге, которая говорит: "об этом подробнее читайте на странице 45". В интернете переход происходит мгновенно.

CSS

Простое объяснение: Это язык, который отвечает за **внешний вид** сайта. Если HTML создает каркас (где заголовок, где текст), то CSS делает этот каркас красивым: задает цвет, размер шрифта, отступы и расположение элементов.

Жизненная аналогия: Если HTML — это стены и перегородки в новом доме (структура), то CSS — это ремонт: поклейка обоев, покраска стен, расстановка мебели (оформление).

Т

Тег

Простое объяснение: Это основная команда или кирпичик в языке HTML. Теги заключаются в угловые скобки <> и говорят браузеру, что именно находится на странице: заголовок, абзац, картинка и т.д.

Жизненная аналогия: Представьте, что Вы редактор и делаете пометки в рукописи. Вы подчеркиваете заголовок и пишете на полях "заголовок" (это тег `<h1>`), отмечаете начало нового абзаца (тег `<p>`). Браузер читает эти пометки и правильно оформляет текст.

Пример: `<p>` — тег для абзаца, `` — тег для изображения.

Ф

Файл

Простое объяснение: Электронный документ или "единица информации" на компьютере. У каждого файла есть имя и расширение (три буквы после точки), которое указывает на тип файла.

Примеры:

`stranica.html` — файл веб-страницы.

`foto.jpg` — файл изображения.

`style.css` — файл со стилями CSS.

Фронтенд (Frontend)

Простое объяснение: Это всё, что пользователь видит и с чем взаимодействует в браузере: кнопки, текст, меню, анимации. Создается с помощью HTML, CSS и JavaScript.

Жизненная аналогия: Витрина магазина и торговый зал, куда заходят покупатели.

Антоним: Бэкенд (Backend) — это то, что скрыто от пользователя: базы данных, серверы, логика сайта. ("кухня" и склад магазина).

Х

Хостинг

Простое объяснение: Услуга по предоставлению места на **сервере** для хранения файлов Вашего сайта. Чтобы Ваш сайт был доступен в интернете 24/7, его файлы должны лежать на специальном, постоянно работающем компьютере — хостинге.

Жизненная аналогия: Если Вы создали сайт — Вы построили дом. Но чтобы его увидели все, ему нужен земельный участок. Хостинг — это и есть тот самый "участок" в интернете, который Вы арендуете у хостинг-компании.

HTML

Простое объяснение: Это язык **разметки гипертекста**. Главный и основной язык, на котором пишутся веб-страницы. Он не является

языком программирования! Его задача — структурировать содержимое страницы: размечать, где заголовок, где абзац, где картинка.

Жизненная аналогия: Скелет или каркас здания. HTML создает основу, несущую конструкцию будущего сайта.

Ч

Чек-лист для создания своего первого сайта:

Написать код — Создать файлы с расширением .html на своем компьютере.

Купить домен — Выбрать и зарегистрировать красивое название для своего сайта.

Купить хостинг — Арендовать место на сервере для своих файлов.

Загрузить файлы — Перенести созданные HTML-файлы с компьютера на хостинг.

Теперь Ваш сайт будет доступен в интернете по вашему доменному имени!

Помните: Не нужно запоминать всё сразу. Возвращайтесь к этому словарику каждый раз, когда встретите незнакомое слово, и скоро эти термины станут для Вас привычными и понятными.

Приложение В: Что учить дальше? Краткий обзор CSS

Дорогие друзья! Поздравляем Вас с освоением основ HTML. Теперь Вы знаете, как создать каркас сайта — расставить заголовки, абзацы, картинки и списки. Но вы, наверное, заметили, что такие страницы выглядят довольно скучно и простовато.

Вот здесь на сцену выходит **CSS** — наш волшебный инструмент для превращения скромного каркаса в красивый и современный сайт!

Что такое CSS? Проще простого!

HTML — это **скелет** вашего сайта. Кости есть, форма есть, но нет кожи, волос и одежды.

CSS — это **кожа, прическа и костюм** для Вашего сайта. Он отвечает за всю красоту!

CSS (Cascading Style Sheets) переводится как "**Каскадные Таблицы Стилей**". Не пугайтесь этого названия! По сути, это просто набор правил, которые говорят браузеру: "Этому заголовку сделай шрифт красным, а тот абзац размести с правой стороны и поставь вокруг него синюю рамку".

Что конкретно можно делать с помощью CSS?

Давайте посмотрим на конкретные примеры, как CSS может преобразить Вашу HTML-страницу.

1. Работа с текстом

CSS позволяет полностью контролировать внешний вид Вашего текста.

Что можно изменить	Пример правила CSS	Как это выглядит
Цвет текста	color: blue;	Синий текст
Размер шрифта	font-size: 24px;	Крупный текст
Семейство шрифтов	font-family: Arial;	Текст другим шрифтом
Выравнивание	text-align: center;	<div style="text-align: center;">Текст по

		центру</div>
--	--	--------------

Было (только HTML):

```
html  
<p>Это простой абзац текста.</p>
```

Стало (с добавлением CSS):

```
html  
<p style="color: darkgreen; font-size: 20px; font-family: Georgia; text-align: center;">  
    Это элегантный абзац с крупным зеленым шрифтом, выровненный по  
    центру.  
</p>
```

2. Работа с блоками и фоном

Вы можете управлять фоном, рамками и отступами элементов.

Что можно изменить	Пример правила CSS
Фоновый цвет	background-color: lightyellow;
Рамка	border: 2px solid brown;
Внутренний отступ (поля внутри блока)	padding: 20px;
Внешний отступ (поля снаружи блока)	margin: 15px;

Было (только HTML):

```
html  
<div>Новость дня</div>
```

Стало (с добавлением CSS):

```
html  
<div style="background-color: lightyellow; border: 2px solid brown; padding:  
20px; margin: 15px;">  
    Новость дня  
</div>
```

(Этот блок будет выглядеть как выделенная цветная плашка с рамкой и отступами)

3. Самый главный принцип: Разделение обязанностей

Обратите внимание на ключевое отличие:

HTML отвечает за **содержание и структуру** ("что показать")

CSS отвечает за **внешний вид и оформление** ("как это показать")

Это как в театре: HTML — это актеры и их реплики (содержание пьесы), а CSS — это костюмы, грим, декорации и свет (оформление спектакля).

Как "прицепить" CSS к HTML?

Есть три основных способа подключить стили к Вашей странице:

1. Прямо в теге (инлайновый стиль)

Используется атрибут style.

Плюс: Просто и быстро для одного элемента.

Минус: Неудобно, если нужно стилизовать много элементов.

Пример: <p style="color: red;">Красный текст</p>

2. Внутри страницы (в разделе <head>)

Стили пишутся внутри тега <style> в заголовке документа.

Плюс: Удобно для стилей одной конкретной страницы.

Пример:

```
html
<head>
  <style>
    p { color: blue; }
    h1 { text-align: center; }
  </style>
</head>
```

3. В отдельном файле (самый правильный способ!)

Все CSS-правила пишутся в отдельном файле с расширением .css (например, style.css).

Этот файл подключается к HTML-документу с помощью тега <link> в разделе <head>.

Плюс: Один файл стилей можно применить ко всем страницам сайта!

Чтобы изменить цвет всех заголовков на сайте, Вам нужно будет поправить всего одну строчку в одном файле.

Пример подключения:

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

С чего начать изучение CSS? План действий

1. Освойте базовые понятия:

Свойства и значения: свойство: значение; (например, color: red;).

Селекторы: Как сказать браузеру, к КОНКРЕТНОМУ элементу применить стиль (например, селектор по тегу p, по классу .text или по идентификатору #header).

2. Начните с простого:

Поэкспериментируйте с цветами (color, background-color), шрифтами (font-family, font-size) и отступами (margin, padding).

3. Практикуйтесь на своей HTML-странице:

Возьмите страничку, которую Вы уже создали с помощью HTML. Попробуйте изменить цвет заголовков, добавить фон всему сайту, поиграть с рамками вокруг картинок.

4. Используйте подсказки:

Современные редакторы кода (например, VS Code) автоматически предлагают возможные CSS-свойства, когда Вы начинаете печатать.

Где найти простые и понятные уроки по CSS?

Вот проверенные ресурсы, где материал объясняется так же доступно, как и в этом учебнике:

1. HTML Academy (htmlacademy.ru)

Почему подходит пенсионерам: Огромное количество интерактивных курсов и тренажеров. Вам не нужно просто читать — Вы сразу же выполняете задания в браузере, видя результат. Очень наглядно и понятно. Есть бесплатные интенсивные курсы.

2. Stepik.org

Почему подходит пенсионерам: Платформа с курсами от ведущих вузов и компаний. Курсы структурированы, есть пошаговые задания и система достижений. Найдите курс "Веб-разработка для начинающих: HTML и CSS".

3. Канал "Веб-разработка для начинающих" (YouTube)

Почему подходит пенсионерам: Если Вам проще воспринимать информацию на слух и видеть процесс. Найдите видео с названиями "CSS для начинающих за 1 час".

4. Советский, но надежный способ: Книги

В библиотеках часто можно найти старые издания по HTML и CSS. Несмотря на возраст, основы в них изложены очень подробно и системно, без лишней сложности.

Главный совет: Не пытайтесь выучить всё и сразу. Осваивайте CSS маленькими шагами. Сначала научитесь менять цвет текста, потом — размер, потом — добавлять отступы. Каждый новый навык — это большое достижение!

Помните, что создание сайта — это творческий процесс. CSS дает Вам в руки кисть и краски, чтобы раскрасить созданный Вами HTML-каркас. Удачи в этом увлекательном деле

Приложение Г: Ответы к практическим заданиям

Дорогие друзья! Здесь Вы найдете примеры готовых кодов для заданий, которые были предложены в учебнике. Пожалуйста, используйте их **только для самопроверки!** Сначала обязательно попробуйте выполнить задание самостоятельно, и только если возникли трудности или Вы хотите свериться с ответом — заглядывайте сюда.
Не бывает двух одинаковых решений! Ваш код может немного отличаться от нашего, и это нормально. Главное, чтобы он работал правильно.

Задание из Главы 2: Ваша первая HTML-страница

Задание: Создайте файл index.html. Напишите базовую структуру HTML-документа. В теле страницы (**<body>**) разместите главный заголовок **<h1>** с текстом "Добро пожаловать на мою страницу!" и абзац **<p>** с любым приветственным текстом.

Ответ:

```
html
<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="utf-8">
    <title>Моя первая страница</title>
</head>
<body>
    <h1>Добро пожаловать на мою страницу!</h1>
    <p>Здравствуйте! Меня зовут Иван, и это моя первая веб-страница,
которую я создал самостоятельно. Я очень рад поделиться с Вами своим
новым увлечением.</p>
</body>
</html>
```

Что проверить:

Есть ли **<!DOCTYPE html>** в самом начале?

Все ли основные теги (**<html>**, **<head>**, **<body>**) на своих местах и закрыты?

Отображается ли заголовок крупным шрифтом, а текст под ним — как обычный абзац?

Задание из Главы 3: Простая статья

Задание: Создайте страницу, которая будет представлять собой небольшую статью на тему "Мое хобби". Используйте заголовки разных уровней (`<h1>`, `<h2>`), абзацы (`<p>`), ненумерованный список (``) для перечисления преимуществ хобби и теги для выделения текста (``, `<i>`).

Ответ:

```
html
<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="utf-8">
    <title>Мое хобби - Садоводство</title>
</head>
<body>
    <h1>Мое любимое хобби: Садоводство</h1>

    <p>Всем привет! Сегодня я хочу рассказать Вам о своем увлечении, которое приносит мне <b>огромную радость</b> и <i>умиротворение</i> — о садоводстве.</p>

    <h2>Почему я люблю садоводство?</h2>
    <p>Это занятие позволяет мне быть ближе к природе и видеть результаты своего труда.</p>

    <h2>Что я выращиваю?</h2>
    <ul>
        <li>Помидоры</li>
        <li>Огурцы</li>
        <li>Клубнику</li>
        <li>Разные цветы, особенно <b>бархатцы</b> и <i>ромашки</i></li>
    </ul>

    <h2>Совет новичкам</h2>
    <p>Начните с неприхотливых растений! Не бойтесь ошибок — на них мы учимся.</p>
</body>
</html>
```

Что проверить:

Заголовок `<h1>` самый крупный, а `<h2>` — чуть меньше.
Список отображается с точками (маркерами) перед каждым элементом.

Слова "огромную радость" и "бархатцы" выделены жирным, а "умиротворение" и "ромашки" — курсивом.

Задание из Главы 4: Страница с фотоальбомом

Задание: Создайте страницу "Мой фотоальбом". Добавьте несколько изображений (можно использовать картинки из интернета или со своего компьютера). Обязательно укажите атрибут `alt` с описанием для каждой картинки. Разместите под каждой фотографией подпись.

Ответ:

```
html
<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="utf-8">
    <title>Мой фотоальбом</title>
</head>
<body>
    <h1>Мой летний фотоальбом</h1>

    <p>Вот несколько фотографий с моей дачи в этом году.</p>

    <div>
        
        <p>Моя любимая клумба. Как Вам эти цветы?</p>
    </div>

    <div>
        
        <p>Первый урожай! Огурцы и помидоры с собственной грядки.</p>
    </div>

    <div>
        
        <p>А это мой помощник, кот Васька. Он "отдыхает" после тяжелого дня.</p>
    </div>
</body>
</html>
```

Важное примечание: В атрибуте `src` вы должны указать правильный путь к Вашим изображениям. Если картинки лежат в той же папке, что и HTML-файл, достаточно просто имени файла, как в примере. Если картинки в другой папке, путь будет таким: `src="images/цветы.jpg"`.

Задание из Главы 5: Навигационное меню

Задание: Создайте простой сайт из трех страниц: "Главная", "Обо мне" и "Мои увлечения". На каждой странице создайте навигационное меню с ссылками на все три страницы.

Ответ для файла index.html (Главная страница):

```
html
<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="utf-8">
    <title>Главная страница</title>
</head>
<body>
    <!-- Навигационное меню -->
    <nav>
        <a href="index.html">Главная</a> |
        <a href="about.html">Обо мне</a> |
        <a href="hobbies.html">Мои увлечения</a>
    </nav>

    <h1>Добро пожаловать на мой сайт!</h1>
    <p>Здесь Вы найдете информацию обо мне и моих увлечениях.</p>
    <p>Используйте меню выше, чтобы перейти в разные разделы.</p>
</body>
</html>
```

Ответ для файла about.html (Страница "Обо мне"):

```
html
<!DOCTYPE html>
<html lang="ru">
<head>
    <meta charset="utf-8">
    <title>Обо мне</title>
</head>
<body>
    <!-- Навигационное меню (должно быть на КАЖДОЙ странице) -->
```

```
<nav>
  <a href="index.html">Главная</a> |
  <a href="about.html">Обо мне</a> |
  <a href="hobbies.html">Мои увлечения</a>
</nav>

<h1>Обо мне</h1>
<p>Меня зовут Александр, мне 68 лет. Я на пенсии и активно
осваиваю компьютер и интернет.</p>
<p>Живу в Москве, у меня двое взрослых детей и трое внуков.</p>
</body>
</html>
```

Ответ для файла hobbies.html (Страница "Мои увлечения"):

```
html
<!DOCTYPE html>
<html lang="ru">
<head>
  <meta charset="utf-8">
  <title>Мои увлечения</title>
</head>
<body>
  <!-- Навигационное меню -->
  <nav>
    <a href="index.html">Главная</a> |
    <a href="about.html">Обо мне</a> |
    <a href="hobbies.html">Мои увлечения</a>
  </nav>

  <h1>Мои увлечения</h1>
  <p>В свободное время я люблю:</p>
  <ul>
    <li>Заниматься садоводством</li>
    <li>Читать книги</li>
    <li>Осваивать новые технологии</li>
    <li>Путешествовать по историческим местам</li>
  </ul>
</body>
</html>
```

Что проверить:

Все три файла (index.html, about.html, hobbies.html) созданы и лежат в одной папке.

Меню на каждой странице содержит три ссылки.
Ссылки работают: при нажатии на "Обо мне" с главной страницы Вы
переходите на about.html, и т.д.

Общие советы по проверке:

Всегда сохраняйте файл после внесения изменений (Файл ->
Сохранить или Ctrl+S).

Обновляйте страницу в браузере после сохранения, чтобы увидеть
изменения (клавиша F5 или кнопка "Обновить").

Не бойтесь ошибаться! Ошибки в коде — это нормально. Браузер
часто подсказывает, где может быть проблема. Если страница выглядит
не так, как Вы ожидали, внимательно перепроверьте код, особенно
закрывающие теги и кавычки.

Помните: Практика — лучший учитель. Чем больше Вы
экспериментируете и пишете свой собственный код, тем увереннее Вы
будете себя чувствовать. Успехов в Вашем творчестве