

HTML + CSS (16 занятий)

Базовый уровень

1. Введение в HTML и CSS (VSCode, структура HTML-документа, теги и их типы, атрибуты, `<a>`, подключение CSS, синтаксис CSS).
2. Структурные теги и семантика (`<header>`, `<main>`, `<section>`, `<footer>`, `<article>`, `<aside>`, `<div>`, списки `/`, цвета в CSS).
3. Работа с текстом и шрифтами (`<h1>-<h6>`, `<p>`, ``, ``, ``, оформление текста, псевдоклассы `:hover`, `:active`, `:visited`, подключение шрифтов (Google Fonts, `@font-face`), выбор гарнитур и кеглей, `line-height`, `letter-spacing`).
4. Кнопки и основы CSS-бокс-модели (`<button>`, атрибуты, стилизация, идентификаторы, якорные ссылки, `margin`, `padding`, `width`, `height`, единицы измерения `px`, `%`, `em`, `rem`).
5. Git и GitHub (установка, базовые команды: `init`, `add`, `commit`, `push`, `clone`; создание репозитория, загрузка домашних заданий, `pull-request`).

Макетирование и адаптивность

6. Блочная модель + Flexbox (строчные и блочные элементы, `display`, `flex`, `justify-content`, `align-items`, `gap`, вложенные flex-контейнеры).
7. Позиционирование элементов (`static`, `relative`, `absolute`, `fixed`, `sticky`, `z-index`, media queries для адаптива).
8. Формы и модальные окна (`<form>`, `<label>`, `<input>` (типы), `<textarea>`, `<select>`, `<fieldset>`, `<legend>`, простое модальное окно).
9. Практика: создание навигации и "гамбургер-меню" (адаптивное меню, кнопка-бургер, скрытие/появление элементов).

Работа с медиа и графикой

10. Работа с изображениями (растровая графика: `jpe`, `png`, `gif`; свойства `background-image`, `background-size`, `background-position`, `linear-gradient`, `box-shadow`).
11. SVG и встроенные объекты (`<svg>`, иконки, SVG-спрайты, `<iframe>`, различия растровой и векторной графики).

Стилизация и анимации

12. Анимации в CSS (`transition`, `transform`, `animation`, `keyframes`, эффекты при наведении).
13. CSS-методология (BEM, организация CSS, структура проекта).

Инструменты

14. CSS-препроцессоры и Tailwind (SASS: @ mixin, @ include, @ extend; Tailwind: подключение, основные классы).

15. Основы UI/UX и Figma

(знакомство с интерфейсом, создание простого макета, работа с сеткой, экспорт ресурсов).

Финальный проект

16. Итоговый проект (2 страницы)(шрифты, картинки, иконки, flex, media queries для 1920px, 1024px, 768px, 320px, анимации, формы, бургер-меню, модальное окно).

JavaScript (16 занятий)

Базовый уровень

1. Введение в JavaScript (подключение скрипта, let/const, типы данных, арифметика, console.log, document.body.innerHTML, Math).

2. Строки и работа с вводом (методы строк, шаблонные строки, prompt(), alert(), confirm()).

3. Массивы и их методы (push, pop, shift, unshift, map, filter, reduce, find, some, every, includes).

4. Объекты и основы работы с ними (ключ-значение, доступ к свойствам, перебор объектов for...in, вложенные объекты, Object.keys/values/entries).

5. Условия и логика (if/else, switch, логические операторы &&, ||, !, тернарный оператор).

6. Функции и циклы (функции: обычные, стрелочные, анонимные; setInterval, setTimeout; циклы for, while, for...of, for...in).

Работа с DOM

7. Работа с DOM: выбор и стилизация элементов (querySelector, querySelectorAll, element.style, classList.add/remove/toggle,.textContent).

8. Работа с DOM: динамическое создание элементов + события (createElement, append, remove, innerHTML, forEach, <script type="module">, addEventListener, removeEventListener).

9. События и формы (обработка <form>, <input>, события submit, change, input, валидация форм).

Практика и инструменты

10. Работа с Vite и GitHub в команде (инициализация проекта на Vite, структура проекта, git-branch workflow, работа в парах: clone, branch, commit, push, pull request).

Работа с данными

11. Работа с API (часть 1) (fetch, JSON, протокол HTTP/HTTPS, ошибки при загрузке данных, пагинация, бесконечный скролл).
12. Работа с API (часть 2) (async/await, try/catch, loader/spinner).
13. Хранение данных и CRUD (localStorage, sessionStorage, примеры: заметки, корзина товаров, избранное).
14. Axios и JSON-server (мини-API) (установка, настройка, создание фейкового API, работа с axios).

Завершение

15. Обзор фронтенд-мира (React, Vue, Angular, Next.js, Node.js — что можно учить дальше).
16. Финальный проект (галерея изображений с Pixabay API). Основные требования: поиск изображений; галерея + модальное окно; пагинация; спиннер при загрузке; закрытие модалки — крестик, клик вне изображения, клавиша Escape. Дополнительно (по желанию): избранные изображения (localStorage), адаптивный дизайн.

Next.js (16 занятий)

Базовый уровень

1. Знакомство с Next.js, создание первого проекта (что такое Next.js, отличие от «чистого React»; установка: прx create-next-app@latest; обзор структуры проекта; JSX — что это; компоненты; props — передача данных между компонентами).
2. Стилизация и состояние компонента (CSS-модули и глобальные стили; подключение Tailwind CSS; события onClick и onChange; useState — работа с состоянием; превращаем статические страницы в интерактивные: добавление и удаление элементов).
3. Работа со списками и условный рендеринг (рендеринг массивов через map(); ключи элементов key; условный рендеринг через && и тернарный оператор; доработка предыдущего проекта — отметка задачи как выполненной или невыполненной).

Роутинг и страницы

4. Роутинг в Next.js (файловая система и автоматический роутинг; динамические маршруты [id].js; навигация через компонент Link; создание страниц: «Список задач», «О проекте»).

5. Layout и переиспользование компонентов
(создание общего шаблона Layout; передача children; компоненты Header и Footer; подключение Layout ко всем страницам проекта).
6. Формы и работа с input (управляемые компоненты; события onSubmit, onChange, input; базовая валидация; форма добавления новой задачи).

Работа с данными

7. Работа с API (часть 1)(fetch и axios; загрузка данных из JSONPlaceholder; отображение списка постов; обработка ошибок; спиннер при загрузке).
8. Работа с API (часть 2) (async/await; try/catch; loader; вывод сообщений об ошибках пользователю).
9. Статическая генерация (SSG) (getStaticProps; генерация статичных страниц на этапе билда; пример: статичная страница со статьями/постами).
10. Серверный рендеринг (SSR)
(getServerSideProps; отличие от SSG; пример: страница с актуальными данными на каждый запрос).
11. Динамические маршруты и getStaticPaths
(создание страниц по id; генерация путей на этапе билда; пример: отдельная страница поста или товара).
12. Оптимизация изображений и ресурсов
(компонент Image; автоматическая оптимизация картинок; работа с шрифтами; Head — мета-информация, favicon).

Продвинутый уровень

13. API Routes в Next.js (создание серверных функций прямо в Next.js; пример: обработка формы обратной связи; работа с req и res).
14. Аутентификация и работа с пользователем
(подключение NextAuth.js; регистрация и вход; работа с сессиями; ограничение доступа к страницам).
15. Деплой и работа с Vercel
(развёртывание проекта на Vercel; переменные окружения .env; настройка production-билда; проверка производительности Lighthouse).

Финальный проект

16. «Галерея изображений» (поиск картинок через Pixabay API; список картинок в виде сетки; модальное окно с данными об авторе, тегах и лайках; пагинация и бесконечный скролл; спиннер при загрузке; избранное с сохранением в localStorage; закрытие модалки по клику и клавише Escape; адаптивный дизайн под смартфоны).

HTML&CSS

1. Введение в HTML CSS (знакомство с редактором VSCode, что такое HTML, как работает Веб страничка, структура HTML, что такое тег и их типы, открывающийся и закрывающийся теги, пустые или самозакрывающийся теги
, , знакомство с тегом <a> и его атрибутами);
2. Структурные теги: <header>, <main>, <section>, <div> (список и другие типы списков, что такое CSS и подключение стилей, синтаксис написание CSS свойств, типы обозначения цвета);
3. Тег <button> и якорные ссылки (атрибуты кнопки, стилизация через псевдоклассы, идентификатор элемента и якорь кнопки, бокс-модель, свойства определение размеров: px, rem, em, %; свойство прозрачности в цветах, способы задания отступов; одно значение, 2, 3 или 4 значения.);
4. Бокс-модель (флекс-бокс) (блоковый теги, строчные теги, флекс-бокс основные свойства, структурные псевдоклассы):
5. Позиционирование элемента (static, relative, absolute, fixed, sticky, z-index, медиа запросы);
6. “Гамбургер меню” и модальное окно (<form>, <label>, <input> - типы ввода, <select>, <fieldset>, <legend>);
7. Git и GitHub (установка Git, создание и работа с репозиториями в GitHub);
8. Изображение - растревые .jpeg, .gif и .png (свойство background-image, свойства background-repeat, background-size, background-position, background-attachment, псевдоклассы ::before и ::after, linear-gradient, box-shadow);
9. Векторная графика .svg (<svg> и его свойства, SVG - sprite, <iframe>);
10. Анимация (CSS переходы, transition-property, transition-duration, transition-timing-function, transition-delay);
11. CSS методология (BEM, домен, хостинг);
12. CSS препроцессор (SASS, @ mixin, @ include, @ extend, @ each);
13. Библиотека Tailwind CSS (подключение, использование стилей);
14. Основы UI дизайна, инструмент Figma;
- 15-16. Заключительный проект HTML CSS (подключение шрифтов, картинки, иконки iframe, @ mixin, использование flex, медиа запросы на 1920px, 1024px, 768px и 320px, анимация, модальное окно формулар input, button, textarea, минимум 2 страницы).

Java-Script

1. Введение в JS (подключение скрипта, объявление переменных, основные типы данных, арифметические операции, просмотр результата нашего кода с помощью console.log() или document.body.innerHTML, основные методы функции Math);
2. Работа со строкой (основные методы, ввод prompt() и вывод alert());
3. Объекты и массивы (основные методы работы с ними);
4. Условие (операторы условий, логические операторы && и ||, тернарный оператор);
5. DOM (querySelector, element.style., className.add, className.remove, className.toggle, classList,.textContent);
6. DOM (createElement, append, remove, import, <script type=module>, forEach);
7. Функции и циклы (обычные, стрелочные, анонимные, setInterval(), setTimeout(), циклы for, while, for...in, for...of);
8. Слушатель события (addEventListener(), removeEventListener(), типы события);

9. Слушатель события, обработка <form> и <input>;
 10. Работа с API и динамический контент (fetch, загрузка JSON, протокол HTTP (HTTPS), основные ошибки при загрузки данных, пагинация, бесконечный скроллинг);
 11. Работа с API (функции Async / await, обработка запросов через блок try/catch, работа loader);
 12. CRUD (методы для работы с localStorage);
 13. Создание собственного API (использование apidroid или json-server, использование библиотеки, axios);
 14. Заключительное занятие (объяснение что можно изучить дальше, краткое знакомство с фреймворками: React, Angular, Next.js, Node.js, Vue);
 - 15-16. Завершающий проект по JavaScript. Требования к функционалу:
 - Форма поиска изображений — текстовое поле для ввода запроса;
 - После отправки формы выполняется поиск и отображаются результаты;
 - Отображение результатов — изображения отображаются в виде галереи (сетки), каждое изображение кликабельно и при нажатии открывается в модальном окне с подробной информацией: автор, теги, количество загрузок, количество лайков;
 - Пагинация — отображение нескольких страниц результатов с помощью кнопок «Предыдущая» и «Следующая», информация о текущей странице (например, «Страница 1 из 5»);
 - Анимация загрузки (спиннер) — отображается при загрузке данных из API;
 - Закрытие модального окна — нажатие на крестик, нажатие за пределами изображения, нажатие клавиши Escape;
 - Дополнительные задания (необязательно, но рекомендуется):
 - ★ Избранные изображения — каждое изображение имеет кнопку «★», которую можно использовать для добавления в избранное. Избранные изображения можно просмотреть, нажав кнопку «Просмотреть избранное». Классическая пагинация в избранном не отображается. 
 - Адаптивный дизайн — приложение должно хорошо работать на мобильных устройствах (например, смартфонах и планшетах).
- Документация API Pixabay: <https://pixabay.com/api/docs/>.