

студенты

15-55 лет

top

КОМПЬЮТЕРНАЯ
АКАДЕМИЯ



учёба с нуля
и бесплатно



диплом о проф.
переподготовке



трудоустройство
во время обучения



живая помощь
от преподавателей



платформа
с упражнениями



обновленная
программа
2025

Курс

Front-end

736 академических часов

1 год длительность курса

368 часов аудиторных занятий

368 часов самостоятельной работы

Наши партнёры:

Yandex Cloud

hh карьерный маркетплейс

Разработка веб-страниц на языке разметки HTML5 с использованием каскадных таблиц стилей CSS3

Введение в Web-технологии. Структура HTML. Форматирование текста при помощи HTML

- Введение в предмет и языки разметки. Язык разметки гипертекста HTML.
- Структура HTML5 документа.
- Теги — основной элемент структуры HTML. Правила записи тегов и их атрибутов в стандарте HTML5. Синтаксические отличия HTML4, XHTML, HTML5.
- Классификация тегов: линейные и блочные, логическое и физическое форматирование.
- Основные ошибки в записях тегов.
- Кодировки страницы и теги <meta>.
- Модель форматирования текста: заголовки и абзацы. Элементы <p>, <h1>...<h6>. Выравнивание текста в блочных элементах: атрибут align.
- Практика: создание простейшей веб-страницы.

Форматирование с помощью CSS. Списки. CSS отступы и поля

- CSS — каскадные таблицы стилей.
- Сравнение HTML и CSS на примере линейных и блочных тегов.
- Теги без форматирования <div> — блочный, — линейный.
- Использование атрибутов class и id для задания стилей.
- Использование внешних CSS-файлов стилей.
- Дополнительные свойства CSS для форматирования текста: letter-spacing, line-height, text-indent, text-transform, white-space, word-spacing.
- Практика: форматирование текста при помощи CSS.
- Создание списков.
- Создание вложенных списков.
- Форматирование списков с помощью CSS.
- Списки определений: элементы <dl>, <dd>, <dt>.
- Управление отступами: практические примеры использования.

Materialize

- История создания Materialize.
- Цели и задачи Materialize.
- Подключение Materialize.
- Materialize design.
- Система сеток в Materialize.
- Понятие Helper. Виды Helpers.
- Навигация и Bootstrap.
- Использование изображений и видео.
- Скроллинг.
- Практические примеры использования.

Less

- История создания Less.
- Цели и задачи Less.
- Подключение и компиляция Less.
- Переменные.
- Миксины.
- Функции.
- Практические примеры использования.

ЭКЗАМЕН

Разработка клиентских сценариев с использованием JavaScript и библиотеки JQuery

Введение в JavaScript

- Сценарии, выполняемые на стороне клиента.
- JavaScript.
- История создания JavaScript.
- Различия между JavaScript и Java, JScript, ECMAScript.
- Версии JavaScript.
- Понятие Documents Object Model.
- Понятие Browser Object Model.
- Внедрение в HTML документы.
- Редакторы кода JavaScript.
- Тег <noscript>.
- Основы синтаксиса.
- Переменные. Правила именования переменных.
- Типы данных.
- Операторы.
- Ввод/вывод данных. Диалоговые окна.
- Условия.
- Циклы.
- Функция.
- Объект arguments.
- Область видимости переменной.
- Ключевое this.
- Рекурсия.

Введение в объектно-ориентированное программирование

- Объекты. Массивы. Строки.
- Задержки и интервалы. Периодический вызов функций.
- Объект Date. Обработка даты и времени.
- Объект Math. Свойства и методы. Случайные числа.
- ООП. Три фундаментальных принципа.
- Понятие класса и объекта в терминах JavaScript.
- Свойства.
- Методы.
- Свойства-аксессуары.
- Конструктор.
- Понятие prototype.
- Наследование.

Обработка событий

- Событие. Обработчик события.
- Обработка событий в сценариях.
- Управление стилями элементов веб-страницы.
- Объект event и его свойства.
- Обработчики событий по умолчанию (стандартные обработчики). Запрет вызова стандартного обработчика.
- Объект image. Управление рисунками и ролловерами.

Browser Object Model. Documents Object Model

- Отличия DOM от BOM.
- Объекты Browser Object Model.
- Documents Object Model.
- Представление HTML — документа в виде дерева.
- Объекты модели DOM. Иерархия узлов.
- Модель событий DOM.
- Свойства и методы модели DOM.
- Изменение дерева DOM.
- Знакомство с объектами Document и Link.
- Управление выделением и текстовым диапазоном: объекты Selection и textRange.
- Особенности DOM в HTML5.

Взаимодействие с DOM

- Создание новых элементов DOM.
- Вставка элементов DOM.
- Передвижение элементов DOM.
- Копирование элементов DOM.
- Взаимодействие с атрибутами.

Формы

- Применение форм. Размещение элементов формы в HTML.
- Коллекция Forms.
- Создание и программирование элементов формы.

Проверка достоверности форм. Использование Cookie

- Проверка достоверности данных формы.
- Объект RegExp. Правила записи регулярных выражений.
- Методы объектов String и RegExp для работы с регулярными выражениями
- Понятие Cookie.
- Создание, использование и удаление Cookie.
- Преимущества и недостатки Cookie.

Рисование с помощью Canvas, поддержка медиа-возможностей

- Понятие Canvas и ее базовые возможности.
- Поддержка медиа-возможностей.

JSON, Ajax

- Цели и задачи. Синтаксис JSON.
- Объект JSON.
- Настройка пользовательской сериализации в JSON. Метод toJSON.
- Синхронные и асинхронные запросы.
- Ajax.
- Объект XMLHttpRequest.
- Методы и свойства XMLHttpRequest.
- Понятие HTTP заголовка.
- Использование метода GET.URL кодирование.
- Использование метода POST.

ECMAScript 6

- История возникновения.
- Переменные.
- Тип данных Symbol.
- Функции-стрелки.
- Использование строк.
- Объекты и классы.
- Модули.
- Другие возможности ECMAScript 6.

Модульное тестирование

- Цели и задачи модульного тестирования.
- Необходимость модульного тестирования.
- Обзор инструментов для модульного тестирования.
- Практические примеры использования.

Паттерны проектирования

- Понятие паттерна проектирования.
- Причины возникновения паттернов проектирования.
- Принципы применения паттернов проектирования.
- Принципы выбора паттернов проектирования.
- Принципы разделения паттернов на категории.
- Введение в UML.
- Использование UML при анализе паттернов проектирования.
- Порождающие паттерны.
- Структурные паттерны.
- Паттерны поведения.

Паттерн MVC

- Цели и задачи паттерна Model-View-Controller.
- Model.
- View.
- Controller.
- Примеры использования паттерна MVC.

Принципы проектирования классов SOLID

- Принципы проектирования классов SOLID.
- Обзор проблем, встречающихся при проектировании и разработке классов.
- Примеры использования принципов SOLID.

Введение в jQuery

- Цели и задачи jQuery.
- История создания jQuery.
- Версии jQuery.
- Подключение jQuery.
- Доступ к элементам страницы при помощи функции \$.
- Понятие селектора.
- Типы селекторов.
- Traversing.
- Методы обхода DOM: filter, next, nextAll, prev, prevAll, siblings и др.

События и jQuery

- Создание обработчиков событий с использованием jQuery.
- Удаление обработчиков событий.
- Объект Event и jQuery.
- Воздействие на обработку события.
- Запуск обработки события.

Стили и анимация

- Метод CSS.
- Отображение и скрытие элементов с помощью методов show и hide.
- Создание эффектов.
- Анимация.

Ajax и JQuery

- Понятие JSON.
- Механизмы Ajax внутри библиотеки JQuery.
- Использование метода GET.
- Использование метода POST.
- События и Ajax в рамках JQuery.
- Обработка ошибок.

Использование JQuery плагинов

- Понятие плагина JQuery.
- Подключение плагина.
- Примеры плагинов.

Работа в команде, управление программными проектами

- Управление. Причины возникновения дисциплинарных проблем во время управления программными проектами.
- Диаграммы Ганта.
- Важные вопросы по управлению программными проектами.
- Анализ терминов предметной области.
- Характеристики проекта.
- Расходы, связанные с проектом.
- Общий обзор моделей и методологий процесса разработки.
- Подробнее о Scrum.

Использование фреймворков Angular, React

NodeJS

- Цели и задачи NodeJS.
- Настройка окружения.
- Инструменты для разработки и отладки.
- Структура приложения.
- Менеджер пакетов npm.
- Создание и регистрация собственных пакетов.
- Работа с событиями.
- Использование файловой системы.
- Модули по взаимодействию с HTTP и URL.

Сборка проектов с помощью Webpack 2 и других инструментов

- Сборка проектов.
- Проблемы, возникающие при сборке проекта.
- Инструменты для сборки проектов.
- Webpack 2.

Angular

- Цели и задачи Angular.
- Понятие одностраничного приложения (SPA).
- Анатомия Angular приложения.
- Настройка окружения.
- Понятие модуля.
- Компонент.
- Шаблоны.
- Интерполяция.
- Использование директив.
- Data binding в Angular.
- Pipes в Angular приложении.
- Сервисы.
- Dependency Injection.
- Использование HTTP запросов.
- Навигация и пересылка данных.
- Практические примеры использования.

React

- Цели и задачи React.
- Анатомия React приложения.
- Настройка окружения.
- Понятие Flux.
- Понятие JSX.
- Использование компонентов.
- Жизненный цикл приложения на React.
- Props и State.
- Controller views.
- Понятие Route
- Использование форм.
- Использование Flux.
- Практические примеры использования.

ЭКЗАМЕН

ЭКЗАМЕН

ПО ОКОНЧАНИИ КУРСА СЛУШАТЕЛЬ БУДЕТ:

- Владеть основами взаимодействия с NodeJS. Производить сборку проекта.
- Разбираться в тонкостях использования Webpack.
- Понимать структуру Angular приложения.
- Применять правильные конструкции Angular в зависимости от поставленной задачи.
- Производить связывание данных в Angular приложении. Применять Dependency Injection.
- Разбираться в тонкостях реализации React приложений.
- Использовать Flux.
- Взаимодействовать с формами. Применять основы HTML: теги, атрибуты и способы.
- Структурирования содержимого web-страниц для создания форматированных документов.
- Применять основы CSS: значения, списки, цвета, шрифты и другие метрики форматирования.
- Владеть навыками проверки и отладки кода web-документов.
- Формировать содержимое «web-документов разной сложности.
- Быстро и качественно форматировать сложные web-документы.
- Работать с базовыми конструкциями языка JavaScript такими как переменные, условия, циклы, массивы функции, и т. д.
- Применять ООП и его основные понятия. Исправлять возникающие ошибки.
- Разбираться в понятиях события, обработчика события. Создавать функции-обработчики различных событий.
- Понимать отличия BOM и DOM. Понимать отличия и взаимодействовать с объектами из BOM и DOM. Разбираться в тонкостях реализации клиентских сценариев в разных браузерах.
- Владеть принципами создания форм и анализа данных пользователя с использованием регулярных выражений.
- Сохранять пользовательские данные с помощью механизма cookies.
- Понимать особенности применения HTML5 по отношению к JavaScript.
- Сериализовать и парсить данные используя JSON.
- Владеть принципами создания асинхронных запросов при помощи Ajax.
- Разбираться в паттернах проектирования. Использовать системы контроля версий.
- Владеть принципами командной работы. Выполнять модульное тестирование.



17

лет на рынке



380

филиалов в России



2 800

преподавателей



120 000+

студентов



24 000+

ОТЗЫВОВ



4,8

рейтинг
на Я. Картах



Также доступно
онлайн-обучение

*данные действительны
на январь 2025 года