

Занятие 1. Javascript. Базовые возможности

Темы:

- Основы JavaScript
- Качество кода
- Структуры данных
- Замыкания, область видимости
- Методы объектов и контекст вызова
- Некоторые другие возможности

Практика:

Прорешивать задания из уроков с сайта <https://learn.javascript.ru>

Занятие 2. Javascript. ООП. ES6

Темы:

- ООП в функциональном стиле
- ООП в прототипном стиле
- Современные возможности ES-2015 (ES6)

Практика:

Прорешивать задания из уроков с сайта <https://learn.javascript.ru>

Занятие 3. Vue. Основы

Темы:

- Введение
- Экземпляр Vue
- Синтаксис шаблонов
- Вычисляемые свойства и слежение
- Работа с классами и стилями

Практика:

1. Сделать форму быстрого перевода из одной единицы измерения в другую по формуле, например:

- футы в метры
- морские мили в километры
- градусы в радианы
- температуру по Цельсию в температуру по Фаренгейту

Значения должны обновляться на лету с помощью вычисляемых свойств.

2. Сделать форму заполнения резюме для поиска работы.

Пользователь заполняет текстовые поля с данными о себе. Ниже на странице располагается шаблон резюме, в который введенные данные подставляются.

Список полей:

- профессия
- город
- ссылка на фото
- фио
- телефон
- email
- дата рождения
- образование
- желаемая зарплата
- ключевые навыки
- о себе

Резюме должно формироваться на лету с помощью вычисляемых свойств.

Здесь и далее для верстки рекомендуем использовать bootstrap.

За основу берите внешний вид резюме на hh.ru, но точное соответствие не требуется.

Занятие 4. Vue. События. Формы. Валидация. Основы компонентов

Темы:

- Условная отрисовка
- Отрисовка списков
- Обработка событий
- Работа с формами
- Валидация
- Основы компонентов

Практика:

1. Доработать форму заполнения резюме из предыдущего урока.

Добавить валидацию номера телефона средствами vue. Телефон может состоять только из цифр и иметь длину от 6 до 10 символов.

Доработать блок “Образование”. Он должен состоять из селекта “Уровень” и доп. полей.

Варианты уровней образования:

- Среднее
- Среднее специальное

- Неоконченное высшее
- Высшее

Если выбран уровень образования “Среднее”, то никаких дополнительных полей не появляется.

Если выбран один из вариантов “Среднее специальное”, “Неоконченное высшее”, “Высшее”, то появляются доп. поля

- Учебное заведение
- Факультет
- Специализация
- Год окончания

Введенные данные подставляются в шаблон резюме при нажатии кнопки “Применить”.

Занятие 5. Vue. Организация разработки, сборщики, полезные инструменты

Темы:

- Однофайловые компоненты
- Vue Loader
- Возможности webpack
- Рекомендации по написанию кода
- Инструменты отладки

Практика

Предыдущее задание оформить в виде однофайловых компонентов (перенести код в .vue файлы). Настроить сборку с помощью vue-loader. Настроить минификацию js-файлов, использовать Babel для совместимости со старыми браузерами.

Занятие 6. Vue. Компоненты. Работа с внешними API

Темы:

- Регистрация компонентов
- Входные параметры
- Пользовательские события
- Слоты
- Обработка крайних случаев
- Работа с внешними API

Практика:

1. Доработать форму заполнения резюме из предыдущего урока.

Организовать код в виде дерева компонентов, как описано в документации

Добавить в форму поле Статус в виде селекта из 4 вариантов:

- Новый
- Назначено собеседование
- Принят
- Отказ

Статус по умолчанию - Новый.

Доработать поле ввода города. Используя апи Вконтакте, выводить выпадающий список подсказок при вводе названия.

<https://vk.com/dev/database.getCities>

Доработать поле ввода названия университета. Используя апи Вконтакте, выводить выпадающий список подсказок при вводе названия (учитывать ранее указанный город)

<https://vk.com/dev/database.getUniversities>

Добавить возможность указывать несколько образований.

Под дополнительными полями (учебное заведение, факультет, специализация, год выпуска) выводится ссылка "Указать еще одно место обучения".

При нажатии выводится новый блок дополнительных полей. Сбоку от блока дополнительных полей выводится иконка с крестиком, при нажатии на нее блок удаляется.

Введенные данные подставляются в шаблон резюме при нажатии кнопки "Применить".

Занятия 7, 8, 9. Vue. Single Page Application

Темы:

- Управление состоянием (vuex)
- Роутинг
- Хранилище (local storage)
- Взаимодействие с бэкендом

Практика:

Реализовать приложение "База данных резюме" с помощью ранее созданной формы.

Приложение должно работать без перезагрузки страницы (single page application).

Резюме должны храниться в базе данных.

На стороне backend нужно реализовать REST API со следующими методами:

- [GET] /api/cv - получение списка резюме
- [GET] /api/cv/{id} - получение резюме по id
- [POST] /api/cv/{id}/add - добавление новой записи
- [POST] /api/cv/{id}/edit - редактирование записи

- [POST] /api/cv/{id}/status/update - обновление статуса резюме

Методы должны отдавать и получать данные в формате JSON.

Рекомендуем в качестве СУБД использовать mongodb. Язык программирования на стороне backend - на ваше усмотрение.

Примечание: в данном задании реализовывать пагинацию в методе списка резюме не нужно (для упрощения задачи). В боевых задачах при выводе списков нужно предусматривать пагинацию или ставить ограничение по количеству записей.

Приложение состоит из 3 страниц:

Главная страница /

В верхней части выводится кнопка “Новое резюме. При клике на кнопку открывается форма добавления резюме по адресу /add.

Ниже выводится список резюме из базы данных, разбитый на 4 колонки (по колонке для каждого статуса). В заголовке колонки выводится название статуса и кол-во резюме в этом статусе:

Новый (4) | Назначено собеседование (6) | Принят (2) | Отказ (18)

В списке у каждого резюме выводятся поля

- фото
- профессия
- ФИО
- возраст

Резюме можно перетаскивать между колонками, при этом должен меняться статус в базе данных, обновляться кол-во в заголовке колонки.

Для этой части рекомендуем пользоваться возможностями vueх. Drag-and-drop можно сделать с помощью плагина <https://github.com/SortableJS/Vue.Draggable>.

При клике на резюме в списке открывается страница редактирования по адресу /edit/{id}.

Форма добавления резюме /add

Поведение аналогично форме из предыдущих заданий. Заполненные данные должны сохраняться в бд при нажатии кнопки “Применить”.

Форма редактирования резюме /edit/{id}

Аналогично форме добавления, но в форму подтягиваются данные из бд. Если в адресной строке указан несуществующий id, выводим сообщение об ошибке. При нажатии на кнопку “Применить” данные сохраняются в бд.

Формы добавления и редактирования должны в том числе открываться по прямой ссылке.