

FUNBOX

Квалификационные задания
для разработчиков JavaScript



О заданиях

Данные задания были разработаны для того, чтобы мы могли оценить уровень ваших знаний. Они составлены таким образом, что вы можете выполнить их в любое удобное для вас время в обстановке, которую считаете оптимальной.

Задания разделены на 2 категории.

Level 1 содержит вопросы, которые помогут вам продемонстрировать свой опыт и знание теоретических основ JavaScript.

Level 2 содержит задание, которое поможет вам продемонстрировать свои навыки на практике.

Результат присылайте нам вместе с рассказом о себе на почтовый ящик wanted@fun-box.ru с пометкой «Вакансия разработчика JavaScript». После проверки заданий мы обязательно сообщим вам о нашем решении.

Задания находятся на следующей странице.



Level I

Здесь несколько теоретических вопросов, с помощью которых вы сможете в свободной форме рассказать нам о своем личном опыте в том или ином аспекте, а также проиллюстрировать рассказ примерами своих работ.

Q1

В чем разница между `==` и `===` в JS?

Q2

Зачем подключение внешних скриптов переносят из тега `head` в конец тега `body`?

Q3

Как устроено наследование в JS? Расскажите о своем опыте реализации JS-наследования.

Q4

Опишите различие использования `scope` и синтаксиса «`Controller as`» в `Angular.js`.

Q5

Когда может понадобиться ручной вызов `$digest` в `Angular.js`? Приходилось ли вам его использовать?

Q6

Какие библиотеки можно использовать для написания тестов `end-to-end` в `Angular.js`? Расскажите о своем опыте тестирования веб-приложений.



Level II

На этом уровне находится практическое задание.

Task Description

Необходимо реализовать редактор маршрутов — одностраничное приложение, в котором пользователь в интерактивном режиме может создавать на карте маршрут, указывая начальную, конечную и промежуточные точки движения. Для каждой точки маршрута можно посмотреть ее адрес.

Приложение визуально состоит из:

- текстового поля ввода для новых точек маршрута;
- списка уже введенных точек маршрута;
- интерактивной карты.

На картинке ниже показан схематичный пример, итоговый внешний вид может быть любым, на ваше усмотрение. Приложение должно адекватно вести себя при изменении размера окна браузера, взаимное расположение элементов не должно изменяться.



На следующей странице находятся функциональные и технические требования к заданию.



Task Functional Requirements

Новая точка маршрута добавляется с помощью ввода ее названия в текстовом поле и нажатия Enter. После этого:

- введенная точка маршрута отображается в конце списка уже добавленных точек;
- в центре карты появляется маркер, обозначающий точку маршрута.

Напротив каждой точки маршрута в списке находится кнопка удаления, при ее нажатии точка маршрута пропадает из списка, а с карты пропадает ее маркер.

Порядок точек маршрута в списке можно изменять перетаскиванием.

Маркеры, соответствующие точкам маршрута, можно перемещать по карте перетаскиванием.

Маркеры на карте соединены прямыми линиями в том порядке, в котором они находятся в списке. Полученная таким образом ломаная изображает маршрут, первая точка в списке — начало маршрута, последняя — конец маршрута.

При изменении порядка точек в списке или их удалении, а также при перемещении маркеров маршрут на карте автоматически перерисовывается.

При клике на маркер появляется балун, в балуне отображается название соответствующей ему точки.

Task Technical Requirements

- Яндекс.Карты / Google Maps;
- Angular.js;
- Покрытие полученного кода тестами;
- Небольшое описание того, как смотреть результат (запуск таска, открытие итоговой HTML-страницы в браузере и так далее).

Благодарим вас за время, потраченное на выполнение заданий.