

Тестовые задания:

Задание (1).

Есть форма с набором полей ввода, которые необходимо проверить перед отправкой. Некоторые поля обязательны для заполнения, некоторые должны содержать валидные данные. В качестве примера можно взять форму, приведённую на странице

http://company.yandex.ru/job/vacancies/js_developer.xml

(«Расскажите нам о себе»).

Нужно:

- Сделать удобную функцию для навешивания событий
- С использованием этой функции реализовать следующее:
 - Считать, что обязательны для заполнения поля ФИО, Телефон, E-mail;
 - Поля Телефон и E-mail должны дополнительно проверяться на формат данных (что именно считать валидным телефоном и e-mail, выберите самостоятельно);
 - Вывести информационное сообщение в случае, если проверка не прошла.

Детали реализации -- на ваш выбор. Не используйте JS-фреймворки в работе, напишите всё на "чистом" JavaScript.

Совместимость: IE 6-7-8, FF 2-3-4, Opera 9-10-11, Chrome, Safari.

Задание (2).

Известно, что почти все браузеры позволяют отключить загрузку графики.

Напишите решение на JavaScript, которое позволит определить, что у пользователя в браузере отключены картинки.

Напишите, в каких случаях это решение может не работать, оцените его надёжность.

Совместимость: IE 6-7-8, FF 2-3-4, Opera 9-10-11, Chrome, Safari.

Можно использовать JS-фреймворк на ваш выбор, кроме специальных плагинов, реализующих указанную проверку «из коробки»

Задание (3):

Каждый день на Яндексе регистрируются десятки тысяч человек.

Страница регистрации:

<https://passport.yandex.ru/passport?mode=register>

-- это точка входа для всех новых пользователей.

Оцените процесс регистрации с точки зрения разработчика интерфейсов, составьте его обзор в свободной форме. Что, по-вашему, можно улучшить или исправить? Представьте, что вы отвечаете за разработку этого интерфейса. Если перед вами поставят задачу поднять конверсию регистрации (т.е. увеличить процент пользователей, успешно завершивших регистрацию), как вы спланируете свою работу?

Ответы

Задание (1).

Функция для клиентской валидации.

Готовая страница находится на <http://i-novak.appspot.com/>

Код: на https://github.com/ignovak/my_sandbox

В общем, поскольку задание довольно примитивное, функция тоже простая. Тем не менее, есть возможность для масштабирования и добавления новых фич (в коде оставлены комментарии в тех местах, где можно что-то улучшить или допилить)

Задание (2).

Решение на JavaScript, которое позволит определить, что у пользователя в браузере отключены картинки.

Первый вариант. Добавление скрытого элемента `img` (за пределами страницы) и обработчика `onload` для него. Если картинки включены, обработчик будет вызван, в противном случае - нет. Недостаток: не работает в Safari, нужна какая-нибудь дополнительная проверка (я сделал `img.offsetWidth > 0` ?)

Решение реализовано здесь:

<http://i-novak.appspot.com/static/check-images1.html>

Второй вариант (fail: не работает в firefox). У первого метода есть недостаток: требуется загрузить реальную картинку. Если в силу каких-либо причин она не загрузится, ответ будет некорректным. Решение: использовать схему `data:uri` для загрузки картинки. Если браузер поддерживает эту схему, картинка будет грузиться всегда, и, таким образом, ошибки не возникнет.

Решение здесь: <http://i-novak.appspot.com/static/check-images2.html>

Недостаток: не работает в IE6, IE7 (в своем примере я применил для них первый способ). В IE также возможно передать код изображения внутри html-кода, используя протокол `mhtml`, но решение этой задачи реально требует много телодвижений, и в данном конкретном примере я от нее отказался. Тем не менее, при необходимости решение можно реализовать. Здесь несколько наиболее популярных способов: <http://www.google.ru/search?q=ie6+data+uri>.

Как оказалось, этот способ не работает в firefox, так как тот рендерит данные, загруженные через `data:uri` даже при отключенных картинках. Следовательно, для него также нужно применять первый способ, который по-настоящему является универсальным решением.

Задание (3):

Оценка страницы <https://passport.yandex.ru/passport?mode=register>

В первую очередь хочу сказать, что для получения более-менее полной картины нужно собрать мнение нескольких людей, которые могут оценить интерфейс как пользователи. У меня таких ресурсов нет, с точки зрения пользователя у меня есть только мое субъективное мнение. Тем не менее попытаюсь дать некоторую оценку.

Сразу могу отметить приятный дизайн. Интерфейс вполне интуитивен, все элементы управления удобны, лишних (раздражающих) элементов нет. Внешний вид, содержание и поведение всплывающих подсказок, скрытых описаний и советов продуманы достаточно хорошо. Мелкие моменты в UI, которые я заметил:

- возможно, стоит переместить описание логина вниз к соответствующему полю (сейчас оно находится рядом с полем Имя и может кого-то сбить с толку)
- желательно реализовать возможность выбора suggested логина при помощи клавиатуры (табом или стрелками) без использования мыши
- возможно, стоит слегка увеличить размер сообщений об ошибках; также можно добавить больше обработчиков к событиям, информирующим об ошибках (например, `onblur`)
- так как поле Телефон необязательно для заполнения, для последовательности ему также нужно добавить placeholder "необязательно". Или, наоборот, всем остальным полям добавить метку `required` (такое поведение более привычно для пользователя). Btw, placeholder в поле Альтернативный E-mail слегка смещен вверх.
- было бы полезно сделать обновление капчи по клику по ней (не только по ссылке)
- возможно, стоит реализовать разные локализации
- Low priority (здесь о случаях при отключенных `js/images`, т.е. довольно редких случаях, следовательно, имеющих низкий приоритет):
 - при отключенном `js`
 - индикация ошибок сделана на достаточно хорошем уровне, тем не менее, по возможности, ее можно попробовать улучшить
 - некоторые ссылки некликабельны
 - при отключенных изображениях нет предупреждающего сообщения о невозможности завершения регистрации вследствие отключенной капчи (мобильная версия сделана чуть лучше: капча находится на первой странице)

Основной, как по мне, спорный момент: реализация регистрации в два этапа. При возвращении со второй страницы обратно на первую данные как будто теряются (даже если браузер их сохраняет для повторной отправки формы, пользователь этого может не знать). Вообще, многие пользователи не любят проходить процесс регистрации в несколько этапов (поскольку не могут определить, сколько времени он займет). В конкретном случае вопрос, стоит ли объединить две страницы в одну, - довольно спорный, тем не менее, я бы попробовал этот вариант для сравнения. Количество полей позволяет это сделать, и интерфейс от этого испортиться не должен. В крайнем случае, необязательные для заполнения поля можно скрыть.

Производительность. При первой загрузке размер страниц ~130kB, при повторной ~100. Время загрузки (при высокоскоростном соединении): 1-2s, что вполне приемлемо. Тем не менее, учитывая широкую аудиторию ресурса (у многих пользователей может быть медленное соединение), стоит рассмотреть возможность оптимизации. В частности, использовать gzip сжатие, кэширование на стороне клиента и прокси, оптимизировать порядок загрузки файлов, удалить неиспользуемые участки css-кода)

Прочие моменты, важные скорее с точки зрения разработчика, чем дизайнера:

- страница выглядит одинаково хорошо во всех распространенных браузерах
- страница достаточно адекватно реагирует на изменение размера шрифта и ширины экрана: верстка не ломается, основной контент (форма) всегда доступна и удобна для заполнения
- валидаторы html и css выдают ошибки (хоть с точки зрения дизайна это и не критично)

Так как явных недостатков на странице не обнаружено, при необходимости поднять конверсию я бы для начала однозначно постарался собрать feedback от как можно большей группы пользователей и уже на основе его делал те или иные изменения. В первую очередь можно применить так называемое "коридорное тестирование". Если есть возможность, добавить форму обратной связи, чтобы получать информацию от реальных пользователей. Также можно использовать инструмент page-events tracking сервисов Яндекс.Метрика или Google Analytics для более глубокого анализа, в каком месте и по какой причине пользователь покинул страницу. Статистику нужно обрабатывать с учетом клиентских данных пользователя (браузер, географическое положение и т.п.). Тогда можно будет более-менее точно определить, что нужно делать для повышения конверсии: еще более привлекательный UI или оптимизацию.