

1. Создайте веб-страницу (*order.html*), содержащую форму заказа почтовой доставки какого-либо товара, в которой должны быть следующие поля:

Поле с выпадающим списком, содержащим список возможных товаров для заказа (минимум 3 товара);

Текстовые поля:

- индекс;
- дата доставки;
- имя и фамилия заказчика;
- телефон заказчика;
- адрес электронной почты заказчика.

Реализуйте с помощью регулярных выражений следующие проверки вводимых данных в поля формы:

- поле «Индекс» должно быть длиной только 7 цифр и введенные данные должны быть числом, причем первая цифра только тройкой или четверкой;
- формат даты должен быть ДД.ММ.ГГГГ или ДД.ММ.ГГ;
- длина поля имени и фамилии заказчика должна быть от 7 до 40 символов;
- в поле ввода телефона заказчика реализуйте следующую ситуацию: если в телефонном номере восемь цифр, то номер делится дефисом на одну часть из четырёх цифр, и две части из двух (пример 1234-56-78), если семь – отделяется первая цифры, затем через дефис две группы по три цифры (пример 1-234-567), в остальных случаях оставлять номер без изменений. Введённые данные должны быть числом;
- адрес электронной почты должен проверяться на корректность.

В случае неверного заполнения полей формы рядом с полем ввода должно показываться информационное сообщение с описанием ошибки пользователя (*span*).

Внизу страницы создайте кнопку, при нажатии на которую будет осуществляться проверка корректности введённых данных во все поля формы, в случае наличия ошибок выдающая сообщение об ошибке.

(*onblur, onchange, innerHTML, test*)

2. Создайте веб-страницу (*strings.html*), содержащую следующие элементы:

- Текстовое поле с абзацем из случайного художественного произведения. Определите количество букв «в», «н» в этой фразе (можно взять другие буквы) с учётом и без учёта регистра. *indexOf*
 - Кнопку, по нажатию на которую будет показано количество слов во фразе из текстового поля.
 - Второе текстовое поле, содержащее строку из нескольких предложений.
 - `<div>`, который содержит `<p>`, в котором располагается строка из нескольких предложений, к которым будут применяться преобразования из следующих двух пунктов.
 - Кнопку, по нажатию на которую будет выполнено преобразование строки: каждое чётное слово было приведено к верхнему регистру, а каждое нечётное – к нижнему.
 - Кнопку, по нажатию на которую будет выполнено удаление из строки всех четвёртых букв из вашей фамилии (для фамилии «Иванов» – это «н», для «Долгов» – это «г» и т.п.), с использованием метода *replace* объекта *String*.
 - Текстовое поле, которое будет проверять, является ли введённая строка именем файла с расширением jpg, png или gif. (примеры корректных имён файлов: *img712.jpg*, *updated_img442.png*, некорректных имён: *img412.jpg.tmp*, *lection9.doc*). В случае, если имя файла в строке корректно, рядом с полем ввода должно отображаться сообщение зелёного цвета «Имя файла корректно». В случае, если неправильное – красного цвета «Неверное имя файла».
- (*onkeydown, oncut, onpaste*)

3. Используйте для оформления вашей страницы фреймворк Bootstrap

Контрольные вопросы

1. Для чего используются регулярные выражения?



2. Какой символ в регулярных выражениях означает повторение предыдущего символа 0 и более раз? А какой – один и более раз?
3. Опишите принцип работы «запоминающих» скобок в регулярных выражениях.
4. Приведите примеры строк, которые подходят под регулярное выражение `^mission\s\w+\s\d{3}\s[a-g][d-r][f-z]\.`
5. Какие флаги отвечают за глобальный поиск в строке и игнорирование регистра?
6. Как можно объявить регулярное выражение в коде JavaScript?
7. Что обозначают спец символы `$` и `?` в регулярных выражениях?
8. Опишите механизм работы методов `test` и `exec` в JavaScript.