Теория

DOM <https://developer.mozilla.org/ru/docs/DOM/DOM_Reference/%D0%92%D0%B2%D0%B5%D0%B4%D0%B5%D0%BD%D0%B8%D0%B5>

Что такое JavaScript <https://learn.javascript.ru/intro>

События <https://learn.javascript.ru/introduction-browser-events>

jQuery основы <https://webref.ru/dev/jqfundamentals/jquery-basics>

Приоритет css селекторов <https://puzzleweb.ru/css/18_inherit_cascade2.php>

Разобрать что такое генераторы <http://php.net/manual/ru/language.generators.php>

Задача 1 (генераторы).

Написать функцию которая читает построчно файл с использованием генераторов. Таким образом результат вызова функции можно использовать при обходе foreach.

Задача 2 (регулярные выражения).

Написать функцию которая будет принимать строку и в этой строке преобразует несколько пробельных символов (подряд) и символы переноса строк в один пробельный символ.

Задача 3 (файлы, рекурсия).

Написать функцию которая удаляет каталог (каталог может содержать подкаталоги с файлами)

Задача 4. (куки)

Написать скрипт который проверяет включены ли куки у пользователя.

Задача 5. (php + ajax, файлы, jquery, bootstrap css)

Аякс чат.

UI пользователя: поле ввода для user name, поле ввода для сообщения (textarea). Форму застилить при помощи bootstrap css.

Под формой сделать блок с сообщениями, причем последнее сообщение должно быть сверху. Вместе с сообщением выводить информацию о имени пользователя и дате сообщения.

Пример формы - <http://xiper.net/assets/images/forms/default-inputs-ie6.png>

Пример вывода сообщений - <http://www.altme.com/guide/images/chatpanel.png>

Форму (сообщение) отправлять через аякс. Перед отправкой проверить что пользователь ввел имя и сообщение. После отправки формы текстовое поле для сообщения очищается, user name остается.

При помощи JS функции setInterval() слать запросы через аякс на сервер каждые 2 секунды. Если сервер возвращает новые сообщения - то добавлять их в начало списка через jquery метод prepend().

Подсказка 1. Как определять что на сервере есть новые сообщения? Для этого можно хранить время (микросекунды, float) когда сообщение было сохранено в файл. Пример строки (csv через запятую):

1507130927.646,Ivan,Hi people!

И в момент чтения строки файла сравнивать дату создания (float) 1507130927.646 с неким заданным (float) param\_microtime. Причем param\_microtime будет поступать как раз из аякс запроса. Потом выбираем из всех “новых” сообщений максимально время создания - это и будет новое значение для “param\_microtime”.

Если “param\_microtime” не пришло в аякс запросе то считаем что пользователь только “зашел” в чат, поэтому просто выдаем ему последние 50 сообщений.

Пример ответа: {

success: true,

param\_microtime: “1507130927.646”

messages: [

{

“login”: ‘Ivan’,

“datetime”: “04-10-17 18:28:47”,

“message”: “Hi people!”

},

{

“login”: ‘Sergey’,

“datetime”: “04-10-17 18:27:00”,

“message”: “Hey!”

},

]

}

Подсказка 2. Тестировать чат можно с нескольких браузеров.

Подсказка 3. Как преобразовать microtime в удобочитаемый вид? Можно использовать date('d-m-y H:i:s', (float)1507130927.646).