Лабораторная работа №4.
РАБОТА С ОБЪЕКТАМИ. ТАЙМЕРЫ.

Цель работы.

Изучить основы базового языка по работе с объектами и таймерами JavaScript. Получить основные навыки создания и использования объектов для работы с данными в JavaScript.

Задание.

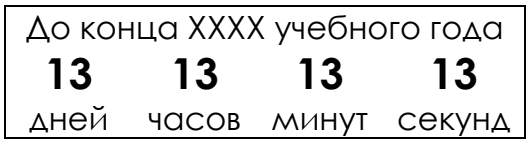
В файле *countries.js* приведён массив *countries* с информацией о государствах мира и массив *about* с заголовочной информацией к массиву *countries* (см. предыдущую работу).

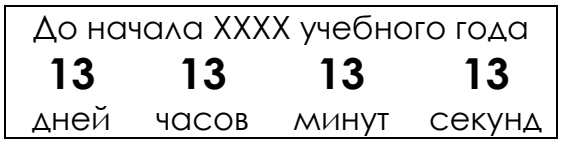
В качестве **индивидуального задания** на основе лабораторной работы №3 необходимо разработать два объекта, предназначенных для отображения и изменения информации о государствах.

Для всех вариантов на панель навигации сайта внедрить блок для отображения времени, оставшегося до:

1. конца учебного года (30 июня) – для нечётных вариантов;
2. начала следующего учебного года (1 сентября) – для чётных вариантов.

Формат отображения времени (XXXX – год, например, 2013):





Порядок выполнения работы.

1. Изучить теоретические сведения по объектам JavaScript.
2. В соответствии с условием индивидуального задания разработать алгоритм изменения данных о государствах (см. *Методические указания* к работе п.1-2). Значение *data* задаётся пользователем.

Вариант	Способ изменения данных	Рекомендации по выполнению
1.	Убрать из географических объектов те, название содержит более *data* символов.	Функция filter() для массивов. *data* – число.
2.	Вывести государств Евразии в указанном регистре.	Функции includes() для массивов и toUpperCase() для строк. *data* – регистр букв: 0 – заглавные буквы, 1 – прописные.
3.	Вывести заглавными буквами названия государств с разницей во времени больше *data* часов.	Функции some() для массивов и toUpperCase() для строк. *data* – число от -10 до 10.
4.	Вывод информации о географических объектах каждого государства в отсортированном виде.	Функция sort() для массивов. *data* – порядок сортировки: 0 – от Я до А, 1 – от А до Я.
5.	Вывести заглавными буквами названия государств, у которых все названия литературных произведений содержат более *data* символов.	Функции every() для массивов и toUpperCase() для строк. *data* – число.
6.	Оставить в списке государств только те, на территории которых расположены Альпы или Карпаты.	Функции includes() для массивов. *data*="Альпы" или "Карпаты"
7.	Убрать из списка литературных произведений те, которые написаны ранее *data* года.	Функция filter() для массивов. *data* – число 1800 до 2010

8.	Вывод информации о литературных произведениях в виде, отсортированном по году выхода издания.	Функция sort() для массивов. *data* – порядок сортировки: 0 – по возрастанию, 1 – по убыванию.
9.	Вывести исторические события буквами в указанном регистре.	Функция toUpperCase() для строк. *data* – регистр букв: 0 – заглавные буквы, 1 – прописные.

3. Продемонстрировать «работу» объектов (см. *Методические указания* к работе п.3)
4. Изучить теоретические сведения по таймерам JavaScript: функции *setTimeout*, *setInterval, clearTimeout* и *clearInterval*.
5. Изучить теоретические сведения по объекту Date.
6. Внедрить счётчик времени на сайт (л.р. №2).
7. Проверить правильность работы разработанных сценариев в различных браузерах.
8. Ответить на контрольные вопросы.
9. Подготовить отчёт по работе.

Методические указания к выполнению.

1. Создать объект *obj1*.

Для первого объекта *obj1* с помощью литерала объектов задать поле-свойство *countries*, которое содержит первоначальную информацию о странах (аналог массива *countries*) и поле-метод *outCountries*, отображающий информацию о странах (см. работу №3):

```
obj1={
   countries: [/*см. работу №3 */],
   outCountries:
      function{
        /* вывод информации о государствах;
         * взять код из работы №3:
         * свояПеременная=countries.map(makeTableFromCountry);
         * свояПеременная.forEach(function(info, i, arr) {
         *       ЭлементДляВывода.innerHTML+=info+"<br>";
         * });
        */
      }
};
```

2. Создать объект *obj2*.

Второй объект *obj2* должен быть создан с помощью функции-конструктора:

```
function Changes() {
   /* */
};
var obj2 = new Changes();
```

Формируемая конструктором структура объекта *obj2*:

```
obj2={
   countries: [/*см. работу №3 */],
   changeCountries:
      function (data){
        /* изменение информации о государствах (по вариантам),
         * data - контрольное значение
```

```
            * для изменения countries согласно заданию
            */
        }
};
```

3. Продемонстрировать «работу» объектов, реализовав следующие действия:
   а). вывести информацию о государствах объекта *obj1* с помощью метода *outCountries*;
   б). вывести имеющуюся информацию о государствах объекта *obj2* с помощью метода *outCountries* объекта *obj1*;
   в). изменить данные *obj1.countries* с помощью метода *changeCountries* объекта *obj2*;
   г). отобразить изменения в браузере с помощью метода *outCountries* объекта *obj1*;
   д). показать как (не)изменились данные поля *countries* объекта *obj2*.

Содержание отчета.

1. Титульный лист.
2. Цель работы.
3. Индивидуальное задание.
4. Описание структуры объектов *obj1* и *obj2*.
5. Описание алгоритмов функций *obj1.outCountries* и *obj2.changeCountries*.
6. Код реализации действий п.3 из *Методических указаний к выполнению*.
7. Программный код создания таймера, а также html-код необходимый для отображения его значения.
8. Выводы по проделанной лабораторной работе.

Контрольные вопросы.

1. Как в коде функции *obj2.changeCountries* получить доступ к *data*, если не указывать это значение в качестве формального параметра функции?
2. К чему приведёт вызов *obj2.changeCountries*, если функция-конструктор формировала бы структуру объекта *obj2* в виде *obj2={changeCountries: function (data){...}};*?
3. В чём разница между вызовами *var o=new Changes()* и *var o=Changes()*?
4. Значение *this* функций-конструкторов, глобальных функций и методов объектов.
5. Различие функций *apply* и *call*.

Приложение. Справочный материал.

1. Инициализация объектов – developer.mozilla.org/ru/docs/Web/JavaScript/Reference/Operators/Object_initializer
2. Работа с объектами – developer.mozilla.org/ru/docs/Web/JavaScript/Guide/Working_with_Objects
3. Функция apply – msdn.microsoft.com/ru-ru/library/4zc42wh1(v=vs.94).aspx
4. Функция call – msdn.microsoft.com/ru-ru/library/h2ak8h2y(v=vs.94).aspx
5. Объект Date – developer.mozilla.org/ru/docs/Web/JavaScript/Reference/Global_Objects/Date
6. Массивы JavaScript – https://developer.mozilla.org/ru/docs/Web/JavaScript/Reference/Global_Objects/Array https://www.w3schools.com/JS/js_arrays.asp
7. Строки JavaScript – developer.mozilla.org/ru/docs/Web/JavaScript/Reference/Global_Objects/String
8. Таймеры JavaScript – www.w3.org/TR/html50/webappapis.html#windowtimers