

Федеральное государственное бюджетное образовательное учреждение  
высшего образования  
«Чувашский государственный университет им. И.Н.Ульянова».  
Кафедра вычислительной техники.  
Предмет: Предмет: WEB-программирование. Основы разработки сайтов.

**Лабораторная работа №3.**  
**Массивы. Динамической вёрстки html-страниц.**  
Вариант 4.

Выполнил: Васильев Егор Юрьевич  
студент группы ИВТ-41-20  
Проверила: Первова Н.В.

**Цель работы:** изучить основы базового языка JavaScript. Научиться создавать JavaScript и внедрять их на web страницы. Получить основные навыки работы с массивами и сданными разных типов, а также навыки динамической вёрстки статических веб-страницы

## Задания

# Италия - Рим

```
tab += '<tr>';
tab += '<td><h1>' + country[0] + ' - ' + country[1] + '</h1></tr>';
tab += '</tr>';
```

Вывод в виде таблицы шириной 75% от доступного пространства для вывода данных. Вывод в виде таблицы в два столбца одинаковой ширины.

*Континент*

Евразия

*Часть света*

Европа

*Разница во времени*

-2

*Денежная единица*

итальянская лира

Вывод в виде аналогичном для «Географических объектов».

*Географические объекты*

1. Альпы
2. Пальмария
3. Апеннинский полуостров
4. Комо
5. Тальяменто

*Литературные произведения*

1. Джованни Верга, Жизнь среди полей, 1880
2. Анна Виванти, Цирцея, 1912
3. Джузеппе Понтиджа, Рождённые дважды, 2000

```
tab += '<tr><td>';
tab += '<table width=75%>';

for (var i = 2; i <= 7; i++) {
    if ( i == 6 || i == 7) {
        tab += '<tr><td><i>' + about[i] + '</i></td><td><ol>' + outArrayData(country[i]) + '</ol></td></tr>';
    } else {
        tab += '<tr><td><i>' + about[i] + '</i></td><td>' + country[i] + '</td></tr>';
    }
}

tab += '</table>';
tab += '</td></tr>';
```

Через цикл формируем таблицу для вывода.

Вывод в виде таблицы, количество столбцов которой зависит от количества исторических дат – полей объекта, содержащего исторические события в country.

## Исторические события

1882 избирательная реформа 1945 антифашистское восстание, 18-28 апреля

```

tab += '<tr><td><b>' + about[8] + '</b></td></tr>';

tab += '<tr><td><table>';

tab += '<tr>' + objectAsTable(country[8]) + '</tr>';

```

## Функция objectAsTable

```

function objectAsTable(historyObject) {
    var historyDates = Object.keys(historyObject);
    var tableLength = historyDates.length;
    var result = "";

    for (i = 0; i < tableLength; i++) {
        var currentKey = historyDates[i]
        result += "<td><b>" + currentKey + "</b> " + historyObject[currentKey] + "</td>";
    }

    return result;
}

```

Достаёт информацию из подаваемого как аргумент объекта и через цикл формирует строку для вывода.

## Функция outArrayData

```

function outArrayData(data) {
    var result = "";

    data.forEach(function (listItem) {
        result += "<li>" + listItem + "</li>";
    });

    return result;
}

```

Преобразовывает элементы массива с названиями географических объектов в строки таблицы.

## Функции, использованные в работе

```

function isNumber(item) {
    return typeof item === 'number';
}

function isTwoArray(item) {
    return Array.isArray(item) && Array.isArray(item[0]);
}

function isArray(item) {
    return Array.isArray(item) && !Array.isArray(item[0]);
}

function isObject(item) {
    return typeof item === 'object';
}

function addToArray(element, item, j) {
    element.push(item);
    element.splice(j, 1);
    return 0;
}

```

**Вспомогательные функции,** которые определяют тип объекта. isNumber определяет, является ли объект числом.

isTwoArray определяет, является ли объект двумерным массивом.

isArray проверяет, является ли объект обычным массивом.

isObject проверяет, является ли объект объектом.

addToArray меняет местами элементы массива

```
return typeof item === 'number';
```

Возвращает строку, указывающую тип операнда

```
arr.forEach(function (element, i, arr) {
```

элементы

Подается массив из которого будут извлечены все

```
var historyDates = Object.keys(historyObject);
```

объектов

Возвращает массив строковых элементов из ключей

```
var tableLength = historyDates.length;
```

данного массива

Возвращает целое положительное число равное размеру

```
table = countries.map(makeTableFromCountry);
```

функции для каждого элемента массива.

Создаёт новый массив с результатом вызова указанной

## Функция сортировки массива

```
function sort(arr) {  
  arr.forEach(function (element, i, arr) {  
    element.forEach(function (item, j, element) {  
      if (isNumber(item)) {  
        addToArray(element, item, j)  
      }  
    });  
    element.forEach(function (item, j, element) {  
      if (isTwoArray(item)) {  
        addToArray(element, item, j)  
      }  
    });  
    element.forEach(function (item, j, element) {  
      if (isArray(item)) {  
        addToArray(element, item, j)  
      }  
    });  
    element.forEach(function (item, j, element) {  
      if (isObject(item)) {  
        addToArray(element, item, j)  
      }  
    });  
    var temp;  
    temp = element[5];  
    element[5] = element[4];  
    element[4] = temp;  
  });  
}
```

Каждый элемент в массиве проверяем на принадлежность к определенным типам. Сначала ищем цифру, т. к она не всегда находится в одинаковой позиции, и переставляем в конец строки, затем меняем местами с денежной единицей. Далее находим двумерный массив и ставим его в конец, находим обычный массив и объект, делаем с ними то же самое.

Сортируем массив и выводим на страницу.

```
sort(countries)

table = countries.map(makeTableFromCountry);

table.forEach(function (info, i, arr) {
  document.body.innerHTML += info + "</br>";
});
```

Функция map вызывается для того, чтобы для каждого элемента массива countries вызвать функцию makeTableFromCountry, и сформировать html массив, который выгружается в тег body.

**Вывод:** изучил теоретические сведения по массивам и типам данным JavaScript. В частности, изучил документацию на функции map и forEach. Практическим путем изучил динамическое построение html документа.