**Лабораторная работа № 5**

**Массивы**

**JavaScript объекты**

* Object(объекты)
* Number (обработка чисел)
* String (обработка строк)
* Array (массивы)
* Math (математические формулы, функции и константы)
* Date (работа с датами и временем)
* RegExp
* Global (его свойства Infinity, NaN, undefined)
* Function

Со всеми данными объектами мы познакомимся позже, с некоторыми из них мы частично уже знакомы.

**Object** — объект, который содержится во всех остальных объектах и обеспечивает их общую функциональность.

**Объект function** — функции являются объектами первого класса, т.е. могут присваиваться переменным, выступать аргументами других функций и быть результатом функций

**Объявление новых объектов происходит одинаково, независимо от объекта:**

*var arr = new Object();  
var str = new String();  
var arr = new Array();*

Как видно из примера, в JavaScript объявление массива происходит точно также, как и у других объектов.

**JavaScript массивы, тип данных Array**

В этой лабораторной работе подробнее рассмотрим объект массив.  
**Массив**— упорядоченный набор данных.

**Доступ к элементам массива** осуществляется с помощью порядкового номера —**индекса**. Таким образом, массив — объект, представляющий собой проиндексированный набор элементов.

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | **var** arr = **new** Array();  arr[0] = "element1";  arr[1] = "element2";  arr[2] = "element3";  alert(arr[2]);  alert("Число элементов" + arr.length); |

В javascript длина массива — свойство length.

**Javascript создание массива**

Создание элементов массива возможно несколькими способами:

|  |  |
| --- | --- |
| 1  2  3  4  5 | **var** earth = **new** Array(4); */\* массив из 4-х элементов\*/*  earth[0] = "Планета";  earth[1] = "24 часа";  earth[2] = 6378;  earth[3] = 365.25; |

|  |
| --- |
| **var** earth = **new** Array("Планета", "24 часа", 6378, 365.25); |

|  |  |
| --- | --- |
| 1  2  3  4  5 | **var** earth = **new** Array(); *// пустой массив*  earth.xtype = "Планета";  earth.xday = "24 часа";  earth.radius = 6378;  earth.period = 365.25; |

|  |
| --- |
| **var** country = ["Россия", "Белоруссия", "Казахстан"]; |

**Javascript: работа с массивами**

Обращение или **доступ к элементам массива в JavaScript** происходит так:

|  |  |
| --- | --- |
| 1  2  3  4 | **var** mas=**new** Array(1,25,'Привет');    mas[0]='Пока';  mas[1]=35; |

**Вывод элементов массива**

Когда мы выводим массив в документ с помощью метода write, то его элементы выводятся через запятую.

1. **Рассмотрим, как осуществляется в javascript вывод массива с использованием обычного цикла for:**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | **var** mas=**new** Array(1,25,'Привет');  mas[0]='Пока';  mas[1]=35;    **function** showElement(){  **for**(i=0;i<3;i++)  alert(mas[i]);  }  showElement(); |

1. **Использование цикла for in для перебора элементов массива:**

|  |  |
| --- | --- |
| 1  2  3  4  5  6 | **function** showElement(){  **for**(**var** i **in** mas){  alert(mas[i]);  }  }  showElement(); |

**Задание 1**. Создать два массива: countries – с названием стран, и population – с населением этих стран. Вывести название страны и ее население (использовать метод document.write)

**Задание 2**. Что выведет на экран следующий фрагмент кода?

|  |
| --- |
| **var** e = 1024;  **var** table = [e, e + 1, e + 2, e + 3];  document.write(table); |

**Пример:** Что выведет на экран следующий фрагмент кода?

|  |  |
| --- | --- |
| 1  2  3  4  5 | **var** arr = [1, 2, 3];  arr[5] = 5;  **for** (**var** i = 0; i < arr.length; i++) {  document.write(arr[i] + "<br />");  } |

В примере в строке arr[5] = 5; происходит расширение массива, в 5-й индекс заносится значение 5. Цикл заполняет элементы типом undefined, элементы которые мы не использовали — они остаются пустыми.

**Удаление элементов массива**  
В Javascript удалить элемент массива можно при помощи оператора delete:

|  |  |
| --- | --- |
| 1  2  3 | **var** myColors = **new** Array("red", "green", "blue");  **delete** myColors[1];  alert(myColors); *// red,,blue* |

**Задание 3**. Создать массив из трех элементов, значения элементов запрашивать у пользователя. Вывести элементы массива на экран, каждый элемент с новой строки

**Свойства массива (Array) в javaScript**

Массив – предопределенный объект. Доступ и обращение к элементам массива осуществляется при помощи точечной нотации (через символ «.»):

***Название\_объекта.свойство\_объекта  
Название\_объекта.метод\_объекта(параметры)***

**Пример:** Вывести элементы массива на экран, в качестве конечного значения счетчика цикла использовать свойство length

|  |  |
| --- | --- |
| 1  2  3  4 | **for** (**var** i = 0; i<arr.length;i++){  alert(arr[i]);  }  alert(arr.length); |

**Поиск элемента в массиве**

В javascript поиск элемента в массиве можно осуществить через функцию:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9  10  11  12  13  14 | **function** findValue(theValue){  **for** (**var** i = 0; i<arr.length;i++){  **if**(arr[i] == theValue)  **break**; *// если нашли - выход из цикла*  }  **if** (i<arr.length) *// если не нашли, то i=arr.length*  **return** i *//если нашли - возвращаем индекс*  **else**  **return** 0; *// если не нашли - возвращаем 0*    }  **var** elem=findValue("element2");  **if** (elem) alert ("элемент найден, его индекс="+elem)  **else** alert("элемент не найден"); |

В примере функция, организованная для поиска элемента массива, имеет один аргумент (theValue) — значение искомого элемента массива. В случае нахождения элемента функция возвращает индекс найденного элемента, если элемент не найден — функция возвратит 0.

**Задание 4**. Создать массив, элементами которого являются названия дней недели. Выводить номер дня недели (номер элемента массива), содержащего введенное пользователем название дня недели (использовать функцию с параметром, которая возвращает значение номера)

**Методы массива (Array) в javaScript**

Рассмотрим некоторые javascript методы массива.

Часто используемым методом в javascript array является **concat().**  
Метод javaScript **concat()** — конкатенация массивов, объединяет два массива в третий массив, т.е. формирует и возвращает объединенный массив, содержащий все элементы исходного массива и значения всех аргументов, переданных методу

***имя\_массива1.concat(массив2)***

**Рассмотрим пример использования метода concat**

|  |  |
| --- | --- |
| 1  2  3  4 | **var** a1= **new** Array(1, 2, "Звезда");  **var** а2 = **new** Array("a", "б", "в", "г");  **var** аЗ = a1.concat(a2);  */\* результат - массив с элементами: 1, 2, "Звезда", "а", "б", "в", "г" \*/* |

**Задание 5.** Дополните код согласно заданию: В переменную a присвоить возвращаемое значение метода concat, передав в этот метод два значения 4 и 5. Что будет выведено на экран?:

|  |  |
| --- | --- |
| 1  2  3 | **var** a = [1, 2, 3];  a = ...  document.write(a); |

Метод javaScript **join**(разделитель) — создает строку из элементов массива с указанным разделителем между ними (преобразует все элементы массива в строки и объединяет их через указанный разделитель)

***имя\_массива.join(строка)***

**Пример использования метода join:**

|  |  |
| --- | --- |
| 1  2  3  4  5 | **var** а = **new** array(1, 2, "Звезда")  a.join(",") *// значение - строка "1,2,Звезда"*    **var** а = **new** array(1, 2, "Звезда")  a.join(" ") *// значение - строка «1 2 Звезда»* |

**Задание 6**. Дополните код, согласно заданию: В строке объявления переменной res присвоить значение: на массиве arr вызвать метод join, в который передать аргумент (тег «br» ) и вывести на экран значение переменной res

|  |  |
| --- | --- |
| 1  2  3 | **var** arr = ["Edward", "Andrey", "Chris"]  **var** res = ...  document.write(res); |

Метод javaScript **shift()** — удаляет первый элемент массива и возвращает результирующий массив, смещая все последующие элементы на одну позицию влево

***имя\_массива.shift()***

**Пример использования метода shift:**

|  |  |
| --- | --- |
| 1  2  3  4 | **var** x = ["a", "b", "c", "d"];  x.shift();  document.write(x);  *//выведет на экран обозревателя строку b,c,d* |

Метод javaScript **рор()** — удаляет последний элемент массива и возвращает результирующий массив

***имя\_массива.рор()***

**Пример использования метода рор:**

|  |  |
| --- | --- |
| 1  2  3  4 | **var** x = ["a", "b", "c", "d"];  x.pop();  document.write(x);  *//выведет на экран обозревателя строку a,b,c.* |

**Задание 7**. Дополните код согласно инструкции:  
Присвоить значение метода pop() в переменную len, который удалит из массива последний элемент:

|  |  |
| --- | --- |
| 1  2  3 | **var** arr = ["January", "February", "March", "April", "May", "June"];  **var** len = ...  document.write(arr.join(" ")); |

Метод javaScript **unshift(значение)** — добавляет к массиву указанное значение в качестве первого элемента

***имя\_массива.unshift(значение)***

**Пример использования метода unshift:**

|  |  |
| --- | --- |
| 1  2  3 | **var** x = ["a", "b", "c", "d"];  document.write(x.unshift("e"));  *//выведет на экран обозревателя число 5* |

Метод javascript **push(значение)** — добавляет к массиву указанное значение в качестве последнего элемента и возвращает новую длину массива

***имя\_массива.push(значение)***

**Пример использования метода push:**

|  |  |
| --- | --- |
| 1  2  3 | **var** x = ['a', 'b', 'c', 'd'];  document.write(x.push('e'));  *//выведет на экран обозревателя число 5* |

Метод javaScript **reverse()** — изменяет порядок следования элементов массива на противоположный

***имя\_массива.reverse()***

**Пример использования метода reverse:**

|  |  |
| --- | --- |
| 1  2  3  4 | **var** x = **new** Array();  x[0] = 0; x[2] = 2; x[4] = 4;  document.write(x.reverse());  *//выведет на экран обозревателя строку 4,,2,,0* |

**Задание 8**. Что выведет на экран следующий фрагмент кода?

|  |  |
| --- | --- |
| 1  2  3 | **var** a = **new** Array(11, 22, 33, 44, 55, 66, 77);  a.reverse();  document.write(a.join("<br/>")); |

Метод javascript **slice(индекс1 [, индекс2])** — создает массив из элементов исходного массива с индексами указанного диапазона (возвращает подмассив указанного массива).

***имя\_массива.slice (индекс1 [, индекс2])***

**Пример:**

|  |  |
| --- | --- |
| 1  2  3  4  5 | **var** a = **new** Array(1, 2, 'Звезда' , 'а' , 'b');  alert(a.slice(1,3));  *// массив с элементами: 2, "Звезда"*  alert(a.slice(2));  *// массив с элементами: "Звезда", "а", “b”* |

**Задание 9**. Дополните код согласно заданию: Присвоить переменной t возвращаемое значение метода slice(), который будет возвращать первых 3 значения из массива:

|  |  |
| --- | --- |
| 1  2  3 | **var** a = [1, 2, 3, 4, 5, 6, 7];  **var** t = ...  document.write(t); |

Метод javaScript **sort()** — сортирует (упорядочивает) элементы массива (если строковые элементы, то сначала произойдет сортировка текста, потом сортировка по алфавиту)

***имя\_массива.sort()***

**Пример использования метода sort():**

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8  9 | **var** arr = [1, 16, 2];  arr.sort();  document.write(arr.join("<p>"));  */\**  *Выведет на экран:*  *1*  *16*  *2*  *\*/* |

Метод javaScript **splice(индекс, количество)** — удаляет из массива несколько элементов и возвращает массив из удаленных элементов или заменяет значения элементов. Т.е. этот метод используется для вставки и удаления элементов из массива.

*имя\_массива.splice(индекс, количество)*

**Пример использования метода splice:**

|  |  |
| --- | --- |
| 1  2  3  4 | **var** a = **new** Array('Ivan','Max','Peter',12,5);  **var** x = a.splice(1,3);  document.write(x+"<br>"); *// Max,Peter,12*  document.write(a); *// Ivan,5* |

**Задание 10**. Дополните код согласно заданию: В переменную d присвоить значение метода splice(), который должен удалить числа 2, 3, 4 из массива:

|  |  |
| --- | --- |
| 1  2  3 | **var** a = [1, 2, 3, 4, 5, 6, 7];  **var** d = ...  document.write(a); |

**toLocaleString(), toString()** — преобразуют содержимое массива в символьную строку

**Задание 11.** Дан массив из чисел: *1, 2, 3, 4, 5*. Распечатать массив в обратном порядке (*5, 4, 3, 2, 1*), используя функцию javaScript reverse()

**Задание 12.** Дан массив из элементов: *‘c’, 5, 2, ‘b’, 3, 1, 4, ‘a’*. Распечатать массив, предварительно отсортировав его по возрастанию (*1, 2, 3, 4, 5, a, b, c*)

**Задание 13.** Дан массив из элементов: *1, 2, 3, 4, 5*. Распечатать исходный массив и массив, получившийся из элементов исходного массива, с разделителем «+» (метод javaScript join – *1+2+3+4+5*)

**Ассоциативный массив в javaScript**

**Ассоциативный массив**— это абстрактный тип данных, по сути, аналог двумерного массива в javascript, позволяющий хранить пары вида «(ключ, значение)». Т.е. ассоциативные массивы позволяют вместо индексов использовать строковые значения.

К сожалению, в javascript не предусмотрены методы для работы с ассоциативными массивами. По этой причине они используются достаточно редко. Однако, их удобно использовать для хранения данных, т.к. их использование облегчает запоминание элементов.

**Пример создания ассоциативного массива:**

|  |
| --- |
| **var** pupil = { name: "Andrey", group: "1" }; |

Другой пример:

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7  8 | **var** m\_list = **new** Object();    m\_list["fat"] = "Полный";  m\_list["small"] = "Маленький";  m\_list["name"] = "Иван";    **for** (**var** x **in** m\_list) *//выведем на экран все элементы*  document.write(m\_list[x] + "<br>"); |

**Задание 14.** Исправить ошибку во фрагменте кода для корректной работы:

|  |  |
| --- | --- |
| 1  2  3  4 | **var** laptop = ( cpu: "Core i7", ram: "4 GB", screen: "15" );  **var** Key = prompt("Введите интересующий параметр (ram, cpu, screen)");  **var** value = laptop[key];  document.write(value); |

**Многомерные массивы**

Так как массивы JavaScript могут в качестве элементов использовать другие массивы, то почему бы эту возможность не использовать для создания многомерных массивов. Для доступа к элементам в массиве массивов достаточно использовать квадратные скобки дважды.

|  |  |
| --- | --- |
| 1  2  3  4  5  6  7 | **var** matrica = [  [5, 5, 5],  [1, 5, 1],  [2, 2, 2]  ];    document.write(matrica[1][1]); *//выбран элемент по центру* |

**Задание 15.** Дана матрица 5 х 5. Вывести элементы матрицы, принадлежащие интервалу[-5; 7]

**Задание 16.** Дана матрица размером N x M. Для каждого столбца матрицы найти произведение его элементов. Обработку массива выполнить с помощью функции.

**Задание 17.** Дана матрица размером N x M. Найти номер последней из ее строк, содержащих только четные числа. Если таких строк нет, то вывести 0. Обработку массива выполнить с помощью функции.