

JS(3) – ОСНОВЫ

Массивы

- Массивы представляют из себя переменные, в которых хранятся наборы значений
- В каждом элементе массива может храниться любой тип
- Индексация начинается с "0"

Массивы(способы задания)

- `var array = [1,3];`//1 способ (РЕКОМЕНДОВАН)

`array[0];` //1

`array[1];` //3

- `var arr = new Array(2,5,9);`// 2ой способ

`arr[0];` //2

`arr[1];` //5

`arr[2];` //9

Размер массива

- `Array.length` – возвращает размер массива.
- Используется для работы с циклами, для вывода значений массива и его обработки
- При уменьшении `length`–массив *обрезается*.
- `arr = [1,2,3];`
- `arr.length; // 3`
- `arr.length=2; // [1,2]`
- `arr.length=3; // [1,2, undefined]`

Копирование

- `var arr=[1,2,5,10,20];`

Array [1, 2, 5, 10, 20]

- `var arr2=arr;`
- `arr2[0]=100;`

Какие значения у массивов теперь?

`arr = ...`

`arr2 = ...`

Копирование идет по ССЫЛКЕ

- `var arr2=arr; (*)`
- `arr` и `arr2` ссылаются на один объект.

- Как создать копию?
- `arr2=arr.slice();`
- `arr2=arr.concat();`

Удаление элементов

- `arr=[1,2,5,10,20];`

`Array [1, 2, 5, 10, 20]`

- НЕПРАВИЛЬНО: `delete arr[3];`

`// Array [1, 2, 5, <1 пустой слот>, 20]`

- ПРАВИЛЬНО `arr.splice(1,3)` //с инд 1 и 3
следующих элем

`//Array [1, 20]`

- Измерьте длину массива в обоих случаях

Методы массива

- `split('Разделитель')` – превращает строку в массив // `'яблоко, мандарин,огурец'.split(',')`
- `join` – обратная операция // `arr.join(',')`
- `slice(start,end)` – Возвращает подмассив от `start` до `end` не включительно
- `splice(start,amount)` – Удаляет `amount` элементов из массива со `start`
- `shift` и `pop` – удаление 1 с начала и конца
- `unshift` и `push` – добавляют с начала и конца

Ассоциативные массивы

- Являются объектами в прямом смысле
- Задаются через точку

```
arr.spb='city'
```

- В стиле массивов

```
arr['spb2']=123
```

```
arr.length - ? // Почему?
```

```
// Сколько выводит элем. при записи "arr"?
```

- Попробуем использовать

```
for (i in arr) console.log('arr['+i+']='+arr[i])
```

Циклы do - while

```
do {  
    // тело цикла  
} while (условие);
```

```
a=0;  
do {  
    a++;  
    console.log('a='+a);  
} while (a<=10)
```

Циклы for стандартный

```
for (начало; условие; шаг) {  
    // ... тело цикла ...  
}
```

```
for (a=1;a<=10;a++){  
    console.log('a='+a);  
}
```

//В чем разница

Циклы for - in

```
for (key in obj) {  
    /* ... делать что-то с obj[key] ... */  
}
```

Вспомним наш чудесный пример:

```
for (i in arr) {  
    console.log('arr['+i+']='+arr[i])  
}
```

Чудо-юдо рыба "let"

- Как мы помним, переменные `var` существуют и до объявления. Они равны `undefined`:

```
alert(a); // undefined
```

```
var a = 5;
```

- С переменными `let` всё проще. До объявления их вообще нет.

Такой доступ приведёт к ошибке:

```
alert(a); // ошибка, нет такой переменной
```

```
let a = 5;
```

Чудо-юдо рыба "let"

Область видимости переменных var и let

Для var – глобальная область видимости

ПРИМЕР - var:

```
var i=10;
```

```
for (;i>3;i--){// можем пропускать части for
```

```
console.log('i='+i);
```

```
}
```

```
console.log('i вне цикла = '+i);
```

Чудо-юдо рыба "let"

Область видимости переменных var и let

Для let - блок

ПРИМЕР - let:

```
var i=10;
```

```
for (let i=5;i>3;i--){
```

```
  console.log('i='+i);
```

```
}
```

```
console.log('i вне цикла = '+i);
```

Различия let и var

- Переменные **var** можно объявлять сколько угодно раз, а переменные **let** только один раз

```
var i=2;
```

```
var i=3; // всё нормально
```

```
let i=2;
```

```
let i=3; // ошибка
```

```
//1 определение на блок
```


Задачи

- Реализовать отображение остатка боеприпасов в пистолете, у которого в обойме N патронов + перезарядку.

Абойма – массив из N ячеек.

Выстрел реализовать через pop/shift

Выстрел или перезарядка – prompt.(пустой)

- * Аналогично реализовать автомат с M патронами в обойме, который стреляет очередью по 3 патрона. Выстрелов не должно быть больше, чем патронов в обойме

Задачи 2

- Заполнить массив заданным количеством городов. (Задается с клавиатуры)

При вводе очередного названия города показывает, сколько раз нужно ввести город

- Заполнить массив произвольным количеством городов. Когда будет введена пустая строка остановить ввод и вывести информацию о городах.

Задачи 3 – поиск простых чисел

Реализовать "Решетó Эратосфéна"

- Выписать подряд все целые числа от двух до n (2, 3, 4, ..., n).
- Пусть переменная r изначально равна двум — первому простому числу.
- Зачеркнуть в списке числа кратные r : $2r, 3r, 4r, \dots$).
- Найти первое незачёркнутое число в списке, большее чем r , и присвоить значению переменной r это число.
- Повторять шаги 3 и 4, пока возможно.

Задачи 4 - циклы

- Составить таблицу значений функции $y = 5 - x^2/2$ на отрезке $[-5; 5]$ с шагом 0.5
- Факториал числа представляет собой произведение всех натуральных чисел от 1 до этого числа... По введенному числу найти его факториал
- Заполнить массив выбранного размера числами фибоначчи

$F_n = F_{(n-1)} + F_{(n-2)}$; //F1=1;F2=1

Задачи 5

- Найти максимальный и минимальный элемент массива и показать его индекс.

Массив заполнить случайными целыми числами

```
Math.floor(Math.random()*(max - min + 1)) + min;
```

- Если успеем решить все задачи выше – дополнительные задачи дам устно.

Дальше консультация

Спасибо за внимание!