

Язык JavaScript

(Дополнительный материал)

ОТСУТСТВИЕ VAR

Переменная, объявленная внутри функции, является локальной по отношению к этой функции и недоступна за ее пределами. Но если непреднамеренно забыть при объявлении переменной `var`, то подразумевается глобальная переменная.

```
function sum(x, y) {  
    // антишаблон: подразумеваемая  
    глобальная переменная  
    result = x + y;  
    return result;  
}
```

Этот фрагмент создаст локальную переменную a, но переменная b будет глобальной.

```
function someFunction() {  
    // антишаблон:  
    var a = b = 0;  
    // ...  
}
```

ШАБЛОН ЕДИНСТВЕННОЙ ИНСТРУКЦИИ VAR

```
function someFunction(){  
    var a = 2,  
        b = 3,  
        sum = a + b,  
        obj = {},  
        i,  
        j;  
    // тело функции  
};
```

ПРИМЕР: ПОМЕНЯТЬ МЕСТАМИ ДВА ЧИСЛА

```
var arr = [1,4,8,2,5,3,7,0,9,6];  
  
for (var i = 0, len = arr.length; i < len-1; i++) {  
    for (var j = 0; j < len-1-i; j++) {  
        if (arr[j+1] < arr[j]) {  
            var tmp = arr[j];  
            arr[j] = arr[j+1];  
            arr[j+1] = tmp;  
        }  
    }  
}
```

ПРИМЕР: ПОМЕНЯТЬ МЕСТАМИ ДВА ЧИСЛА

```
var arr = [1,4,8,2,5,3,7,0,9,6];

for (var i = 0, len = arr.length; i < len-1; i++) {

    for (var j = 0; j < len-1-i; j++) {
        if (arr[j+1] < arr[j]) {
            arr[j] = arr[j] + arr[j+1];
            arr[j+1] = arr[j] - arr[j+1];
            arr[j] = arr[j] - arr[j+1];
        }
    }
}
```

ЗАДАЧА

Сравнить быстроту работы алгоритма сортировки пузырьком целых чисел:

- случай изменения положение двух чисел через промежуточную переменную;
- случай изменения положение двух чисел без промежуточной переменной.

МЕТОДЫ ИТЕРАТОРЫ ДЛЯ РАБОТЫ С МАССИВАМИ

Эти методы предназначены для перебора элементов массива и выполнения различных действий над ними:

имя массива.forEach(callback, thisArg)

```
[2,3,4].forEach(function(el, idx, arr) {  
    document.write('arr['+idx+'] = '+el+' в ['+arr+'] <br>');  
});
```

имя массива.filter(callback, thisObject)

```
var mas = [-1,-2,3,4].filter(function(el, idx, arr){  
    return el<0;  
});
```

имя массива.map(callback, thisObject)

```
var mas2 = [-1,-2,3,4].map(function(el, idx, arr){  
    return el*el;  
});
```


МЕТОДЫ ИТЕРАТОРЫ ДЛЯ РАБОТЫ С МАССИВАМИ

имя массива.every(callback, thisArg)

```
var result = [-2,-3,-4].every(function(el, idx, arr) {  
    return el<0;  
});
```

имя массива.some(callback, thisObject)

```
var result2 = [-1,-2,3,4].some(function(el, idx, arr){  
    return el<0;  
});
```

имя массива.reduce(callback, initialValue)

имя массива.reduceRight(callback, initialValue)

```
var result3 = [1,2,3,4].reduce(function(prev, curr, idx, arr){  
    return prev + curr;  
});
```

ЗАДАЧА

1. Дан массив со строками. Оставьте в нем только те строки, длина которых больше 5-ти символов.
2. Дан массив с числами. Оставьте в нем только положительные числа. Затем извлеките квадратный корень из этих чисел.

МЕТОД ДЛЯ УПРАВЛЕНИЯ СВОЙСТВАМИ ОБЪЕКТА

`Object.defineProperty(obj, prop, descriptor)`

obj - объект, в котором объявляется свойство

prop - имя свойства, которое нужно объявить или модифицировать.

descriptor - объект, который описывает поведение свойства. В нём могут быть следующие поля:

value – значение свойства, по умолчанию `undefined`

writable – значение свойства можно менять, если `true`. По умолчанию `false`.

configurable – если `true`, то свойство можно удалять, а также менять его в дальнейшем при помощи новых вызовов `defineProperty`. По умолчанию `false`.

enumerable – если `true`, то свойство просматривается в цикле `for..in` и методе `Object.keys()`. По умолчанию `false`.

get – функция, которая возвращает значение свойства. По умолчанию `undefined`.

set – функция, которая записывает значение свойства. По умолчанию `undefined`.

МЕТОД ДЛЯ УПРАВЛЕНИЯ СВОЙСТВАМИ ОБЪЕКТА

```
var user = {};
```

```
Object.defineProperty(user, "name",  
  {  
    value: "Саша",  
    configurable: true,  
    writable: true,  
    enumerable: true  
  });
```

```
console.log(user.name);
```

ЗАДАЧА

Создать в объекте свойство-константу.
После чего попробуйте записать в это
свойство значение в обычном режиме
и в режиме 'use strict'.