====================== ТЕОРІЯ ========================

Метод sort() сортує масив. Він сортує в порядку зростання, перетворюючи елементи до рядкового типу. Це дає неправильні результати для числових масивів.

var arr = [1, 2, 15,36,305];

arr.sort ();

alert (arr); / / 1, 15, 2, 305, 36

Методу sort (функція порівняння) вміє сортувати будь масиви, якщо вказати функцію fn від двох елементів, яка вміє порівнювати їх. Вона повинна повертати:

Додатне значення, якщо перший елемент більший,

Від'ємне значення, якщо другий більший,

Якщо рівні – то 0.

function compareNumber (a, b) {

if (a>b) {return 1};

if (a<b) {return -1};

return 0;

//а можна просто return a-b

}

var arr = [1,2,15,36,305];

arr.sort (compareNumber);

alert (arr); / / 1, 2, 15, 36, 305

Приклад. Скласти скрипт для сортування масиву arr порядку спадання

arr.sort(function(a,b){return b-a});

Приклад. Скласти скрипт для сортування масиву arr в порядку, коли спочатку йдуть всі парні чила, потім – непарні.

function evenCompare (a, b) {

  // Числа однієї парності - сортуються звичайним чином

if (a%2 == b%2) return a - b;

// Інакше, якщо a - парне, то воно менше

if (a% 2 == 0) return -1;

// Лишився один варіант: a - непарне, і b – парне

return 1;

}

var arr = [5, 2, 1, -10, 8];

arr.sort (evenCompare);

alert (arr); / / -10, 2, 8, 1, 5

Використання випадкових чисел для функції порівння дозволить перемішати елементи масиву випадковим чином. Math.random() повертає результат що належить [0,1).

var arr = [1, 2, 3, 4, 5];

arr.sort (function () { return Math.random () - 0.5;});

alert (arr); / / елементи у випадковому порядку, наприклад [3,5,1,2,4]

forEach (функція) викликає функцію для кожного елемента масиву. Функція викликається з параметрами (item, i, arr): item – значення поточного елементу масиву; i - його номер; arr - масив, що перебирається. Зміна значення item не приводить до зміни відповідного елементу, для цього слід звертатись до arr[i].

Приклад. Скласти скрипт, що виводить кожен елемент масиву в окремий рядок у форматі a[i]=…

function print(x,i,arr){

document.writeln('a['+i+']='+x+'<br>');

}

function print2(x,i,arr){

document.writeln(x\* x);

}

var a=[2,3,5];

a.forEach(print2);

//АНАЛОГ forEach

for(var i=0;i<a.length;i++)

{

var x=a[i];

print2(x,i,a);

}

alert(a);

filter (callback) створює новий масив, в який увійдуть тільки ті елементи вихідного масиву, для яких виклик callback (item, i, arr) поверне true

Приклад. Створит скрипт, який знаходить усі парні числа

//Функція порівння (визначення чи парне)

function isEven(number, i, arr) {

return number % 2 == 0;

}

//=========================

var a = [2, 11, 56, 43, 23, 890];

var a\_even = a.filter(isEven);

document.write(a\_even);

Приклад. Скласти скрипт, що знаходить всі парні елементи на непарних місцях.

function condition(x,i,arr){

return (i%2!=0)&&(x%2==0);

}

var b=a.filter(condition);

map (callback) створює новий масив, який складатиметься з результатів виклику callback (item, i, arr) для кожного елемента arr.

Приклад. Скласти скрипт, що збільшує кожен елемент масиву на його індекс;

function change(x,i,arr){

return x+i;

}

alert(a.map(change));

every (callback) повертає true, якщо виклик callback поверне true для кожного елемента масива.

some (callback) повертає true, якщо виклик callback поверне true для якогось елемента масива.

Приклад. Чи всі елементи массиву додатні, чи є хоча б один додатній.

var arr = [1, -1, 2, -2, 3];

function isPositive (number) {

return number> 0

}

alert (arr.every (isPositive)); / / false, не всі додатні

alert (arr.some (isPositive)); / / true, є хоч одне додатне

**Приклад. Дано оцінки учня з»ясувати чи є учень двійочником. Якщо не є, то чи є він трійочиником.**

function isNotBad(mark) { //З»ясовуємо чи двійка

return mark>3;

}

function is\_3(mark) { //З»ясовуємо чи трійка

return mark < 7;

}

//=========================

var a = [8, 11, 9, 10, 9];

if (a.every(isNotBad)) { //Перевіряємо чи не двійочник (усі оцінки більші за 3)

if (a.some(is\_3)) { //Перевіряємо чи трійочник (хоча би одна оцінка <3)

alert("3333");

}

else {

alert(">3");

}

}

else {

alert("2222");

}

reduce (reduceCallback [, initialValue]) застосовує функцію reduceCallback по черзі до кожного елементу масиву зліва направо, зберігаючи при цьому проміжний результат.  
Аргументи функції reduceCallback (previousValue, currentItem, index, arr):  
     previousValue - останній результат виклику функції, він же проміжний результат.  
     currentItem - поточний елемент масиву, елементи перебираються по черзі зліва-направо.  
     index - номер поточного елемента.  
     arr - масив.  
Значення previousValue при першому виклику одно initialValue. Якщо initialValue немає, то воно дорівнює першому елементу масиву, а перебір починається з другого.

Функція

*arr*.reduceRight(*callback*[, *initialValue*])

діє аналогічно, але перегляд елементів здійснюється справа-наліво аналогічно.

Приклад. Скласти скрипт для знаходження суми всіх елементів та кількості додатних елементів в масиві.

function kilk(result,x,i,arr){

if (x>0) {result++};

return result;

}

function suma(result,x,i,arr){

result+=x;

return result;

}

var a=[1,2,-3,4,-5];

alert(a.reduce(kilk,0));

alert(a.reduce(suma));

====================== ПРИКЛАДИ =====================

Задача. Створити функцію, яка б могла знаходити суму для довільної кількості переданих чисел

<script>

function sum() { //Усі числа будуть передані у масив arguments

var s = 0;

for (var i = 0; i < arguments.length; i++) {

s += arguments[i];

}

return s;

}

//---------------------------

var s1 = sum(2,3);

var s2 = sum(3, 4, 5);

var s3 = sum(11,2,33,65,23,90);

</script>

---------------------------------------------------------

Задача. Знайти або суму, або добуток чисел.

<script>

function sum(a, b) {

return a + b;

}

function product(a, b) {

return a \* b;

}

//---------------------------

var x = parseInt(prompt("x=", '0'));

var y = parseInt(prompt("y=", '0'));

var answer = parseInt(prompt("1)sum; 2)product", '1'));

var f;

if (answer == 1) //У змінній f зберегти або адресу функції sum або адресу функції product

f = sum;

else

f = product;

var res = f(x, y);

document.write("res="+res);

</script>

Задача. Дано масив цілих чисел. Вибрати одну із дій при перегляді елементів.

Якщо елемент більше за заданий то:

1)помножити на 2;

2)поділити на 7;

Якщо елемент менше рівне за заданий то:

1)помножити на 3;

2)помножити на 5.

function arrayOparations(a,elemetForComparison,f\_greaterThan,f\_lessThan) {

for (var i = 0; i < a.length; i++) {

if (a[i] > elemetForComparison)

a[i] = f\_greaterThan(a[i]);

else

a[i] = f\_lessThan(a[i]);

}

}

//-------- Функції для випадку коли менше

function mult3(number) {

return number \* 3;

}

function mult5(number) {

return number \* 5;

}

//--------- Фунції для випадку коли більше

function mult2(number) {

return number \* 2;

}

function div7(number) {

return number /7;

}

//-------------------------

var a = [2, 11, 5, 6, 23, 87, 66, 54];

var elementForComp = parseInt(prompt("Element", "0"));

//------------------------Визначаємо, що робити з більшими(адресу записуємо у f1\_greater

var answer1 = parseInt(prompt("If greater than : 1) \*2; 2) /7"));

var f1\_greater;

if (answer1 == 1)

f1\_greater = mult2;

else

f1\_greater = div7;

//----------------------- Визначаємо, що робити з меншими(адресу записуємо у f2\_less

var answer2 = parseInt(prompt("If less than : 1) \*3; 2) \*5"));

var f2\_less;

if (answer2 == 1)

f2\_less = mult3;

else

f2\_less = mult5;

//----------------------- Викликаємо узагальнену функцію обробки елементів масиву

arrayOparations(a, elementForComp, f1\_greater, f2\_less);

document.write(a);

//------------------

//function f\_ok() {

// alert("Good boy");

//}

//function f\_bad() {

// alert("We are sorrry ");

//}

//function QustionAnalysis(question, f1, f2) {

// if (confirm(question)) {

// f1();

// }

// else {

// f2();

// }

//}

////--------------------

//QustionAnalysis("Do you like dogs?", f\_ok, f\_bad);

</script>

Задача. Упорядкувати елементи масиву:

1)за зростанням;

2)за спаданням;

3)порівнюючи модулі;

4)порівнюючи квадрати;

5)порівнюючи за останньою цифрою.

<script>

//Задача.Упорядкувати елементи масиву:

//1)за зростанням;

//2)за спаданням;

//3)порівнюючи модулі;

//4)порівнюючи квадрати;

//5)порівнюючи за останньою цифрою.

//------------ Описуємо правила порівняння------------

function Zrost(a, b)

{

return a - b;

}

function Spad(a, b) {

return b-a;

}

function ByModul(a, b) {

return Math.abs(a) - Math.abs(b);

}

function ByKvad(a, b) {

return a\*a - b\*b;

}

function LastDigit(a, b) {

var la = a % 10;

var lb = b % 10;

return la - lb;

}

//--------------- Вибираємо правило порівняння ----

var answer = parseInt(prompt("1.Zrost; 2)Spad; 3)Modul; 4) Kvad; 5)Last digit", '1'));

var rule;

switch (answer) {

case 1: rule = Zrost;

break;

case 2: rule = Spad;

break;

case 3: rule = ByModul;

break;

case 4: rule = ByKvad;

break;

default: rule = LastDigit;

break;

}

var arr = [2, 15, 100, 36, 305];

arr.sort(rule);

alert(arr);

</script>

Задача 1. Дано одновимірний масив. Кожен елемент

1)піднести до квадрату;

2)помножити на 7;

3)поділити на 10.

Використати функції.

//--------- Функції перетворення одного елемента -----

function p1(number) {

return Math.pow(number, 2);

}

function p2(number) {

return number \* 7;

}

function p3(number) {

return number/10;

}

//----------- Загальна функції перегляду усіх елементів з правило перетворення --

function myfunction(a,f\_operation) {

for (var i = 0; i < a.length; i++) {

a[i] = f\_operation(a[i]);

}

}

//=========================

var a = [2, 11, 56, 43, 23, 890];

var answer = parseInt(prompt("1)kv; 2)\*7; 3)/10", "1"));

switch (answer) {

case 1: myfunction(a, p1);

break;

case 2: myfunction(a, p2);

break;

case 3: myfunction(a, p3);

break;

default:

}

document.write(a);

//---------------- З використанням стандартного підходу ------------

//--------- Функції перетворення одного елемента -----

function p1(number,i, arr) {

return arr[i]=Math.pow(number, 2);

}

function p2(number,i, arr) {

return arr[i]=number \* 7;

}

function p3(number, i, arr) {

return arr[i]=number/10;

}

//=========================

var a = [2, 11, 56, 43, 23, 890];

var answer = parseInt(prompt("1)kv; 2)\*7; 3)/10", "1"));

switch (answer) {

case 1: a.forEach(p1);

break;

case 2: a.forEach(p1);

break;

case 3: a.forEach(p1);

break;

default:

}

document.write(a);

</script>