

Not quite what you are looking for? You may want to try:

- Setting up a simple log4net log manager class handling multi appenders and auto zip capability
- How to Use Cursors and While loop in SQL Server



highlights off

12,519,962 members (53,849 online)

Devendra Katuke ▾ 357 Sign out



articles Q&A forums lounge

 *

JavaScript Quick Sort Array Proto Type Method



Cy Scott, 18 Jul 2014

CPOL

Rate:

0.00 (No votes)



Is your email address OK? You are signed up for our newsletters but your email address is either unconfirmed, or has not been reconfirmed in a long time. Please [click here to have a confirmation email sent](#) so we can confirm your email address and start sending you newsletters again. Alternatively, you can [update your subscriptions](#).

Introduction

This is a proto type method for arrays in JavaScript. It utilizes an iterative version of the Quick Sort algorithm. More information about the quick sort algorithm can be found [here](#).

Background

I adapted an iterative version of Quick Sort made in C located [here](#). I tested this iterative version and a recursive version and found that the iterative version usually preforms about twice as fast. I also included some additional features that are not included in the vanilla version of JavaScript sort.

JavaScript Quick Sort Library

Hide Shrink ▲ Copy Code

```
(function()
{
    var
        defaultCompare,
        defaultSwap,
        partition,
        quickSort;

    Array.prototype.quickSort = function(compare, swap)
    {
        if (typeof compare !== "function")
        {
            compare = defaultCompare;
        }
        if (typeof swap !== "function")
```

```

    {
        swap = defaultSwap;
    }
    quickSort(this, 0, this.length - 1, compare, swap);
};

defaultCompare = function(value1, value2)
{
    return value1 < value2;
};
defaultSwap = function(array, index1, index2)
{
    var temp = array[index1];
    array[index1] = array[index2];
    array[index2] = temp;
};
partition = function(array, start, stop, compare, swap)
{
    var pivot = array[stop], storeIndex = start - 1;

    while (start < stop)
    {
        if (compare(array[start], pivot))
        {
            storeIndex++;
            swap(array, storeIndex, start);
        }
        start++;
    }
    swap(array, storeIndex + 1, stop);
    return storeIndex + 1;
};
quickSort = function(array, startIndex, stopIndex, compare, swap)
{
    var pivot, stack, start, top;

    stack = new Array(stopIndex - startIndex + 1);
    top = -1;

    stack[++top] = startIndex;
    stack[++top] = stopIndex;

    while (top > -1)
    {
        stopIndex = stack[top--];
        startIndex = stack[top--];

        pivot = partition(array, startIndex, stopIndex, compare, swap);

        if (pivot - 1 > startIndex)
        {
            stack[++top] = startIndex;
            stack[++top] = pivot - 1;
        }

        if (pivot + 1 < stopIndex)
        {
            stack[++top] = pivot + 1;
            stack[++top] = stopIndex;
        }
    }
};

}();

```

Using the library

These examples explain how to use this code.

Hide Shrink ▲ Copy Code

```
//  
//Plain Sort
```

```

//  

var plainSort = [4, 7, 6, 5, 7, 0, 1, 5, 10, 10, 8];  

plainSort.quickSort();  

console.log(JSON.stringify(plainSort));  

//Prints: [0,1,4,5,5,6,7,7,8,10,10]  

//  

//Sort With Custom Compare Function  

//  

var descendingOrder = [4, 7, 6, 5, 7, 0, 1, 5, 10, 10, 8];  

descendingOrder.quickSort(function(a, b)  

{  

    //Descending order compare  

    return a > b;  

});  

console.log(JSON.stringify(descendingOrder));  

//Prints: [10,10,8,7,7,6,5,5,4,1,0]  

//  

//Sort multiple arrays with a custom compare  

//  

var table =  

{  

    "employeeName": ["Jane", "John", "Tom", "Alex", "Mary"],  

    "id": [10, 5, 7, 3, 0]  

};  

table.employeeName.quickSort(function(a, b)  

{  

    //Strings have to be compared using the LocaleCompare function  

    return (a || "").localeCompare(b) < 0;  

},  

function(array, index1, index2)  

{  

    //Sorts both arrays in the 'table' object  

    var temp = array[index1];  

    array[index1] = array[index2];  

    array[index2] = temp;  

  

    temp = table.id[index1];  

    table.id[index1] = table.id[index2];  

    table.id[index2] = temp;  

});  

console.log(JSON.stringify(table));  

//Prints:{"employeeName":["Alex","Jane","John","Mary","Tom"],"id":[3,10,5,0,7]}

```

License

This article, along with any associated source code and files, is licensed under [The Code Project Open License \(CPOL\)](#)

Share

[EMAIL](#)

[TWITTER](#)

About the Author



Cy Scott No Biography provided
United States

Comments and Discussions

Add a Comment or Question



Search Comments

Go

-- There are no messages in this forum --

[Permalink](#) | [Advertise](#) | [Privacy](#) | [Terms of Use](#) | [Mobile](#)
Web02 | 2.8.161005.4 | Last Updated 18 Jul 2014

Select Language ▾

Layout: [fixed](#) | [fluid](#)

Article Copyright 2014 by Cy Scott

Everything else Copyright © [CodeProject](#), 1999-2016