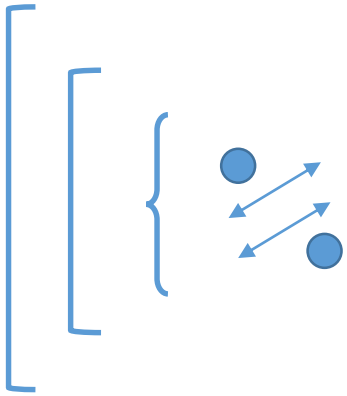
به نام خدا

مرتب سازی حبابی (Bubble Sort) در Java Script

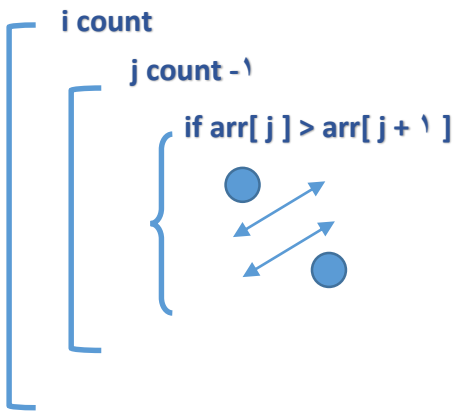
فهرست مطالب

۱.تصویر الگوریتم

این تصویر را به خاطر بسپارید :



حال کمی تصویر را به دنیای برنامه نویسی نزدیک تر می کنیم :



دو حلقه تکرار ، حلقه تکرار دوم (j) یک مرتبه کمتر از حلقه تکرار اول (i) پیمایش می کند.در داخل حلقه تکرار دوم یک دستور شرطی وجود دارد که یک خانه از آرایه را با خانه بعدی آن مقایسه می کند.در صورتی که مقدار خانه از خانه بعدی کمتر باشد عمل جابجایی صورت می گیرد.

این کار تا آخرین خانه آرایه صورت می گیرد و به این ترتیب مرتب سازی انجام می شود که به آن مرتب سازی حبابی می گوییم. در مرتب سازی حبابی اعدادی که کمتر هستند به خانه ی ۰ (بالاتر) نزدیک می شوند. شما می توانید با تغییر علامت بزرگتر یا کوچکتر در داخل دستور شرطی if مرتب سازی را از کوچک به بزرگ یا بلعکس انجام دهید.

۲. تگ Java Script در صفحات وب

در این آموزش می توانید از محیط Visual Studio و یا حتی یک فایل متنی ساده که با NotePad ایجاد شده استفاده کنید. در صورتی که از NotePad استفاده می کنید پسوند فایل را برای اجرا در مرورگر به html تبدیل کنید.

در هر صورت اگر در محیط Visual Studio یک پروژه جدید و یک فایل html ایجاد نمائید متن زیر نمایش داده می شود :

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
```

```
<head>
    <title></title>
</head>
<body>


</body>
</html>
```

به صورت پیش فرض در یک صفحه html تگ Java Script وجود ندارد ، شما می توانید این تگ را در داخل تگ head به این صورت اضافه کنید :

```
    <script type="text/javascript">
    </script>
```

به این ترتیب فایل html به این صورت تبدیل می شود :

```
<!DOCTYPE html>
<html xmlns="http://www.w3.org/1999/xhtml">
<head>
    <title></title>
    <script type="text/javascript">
    </script>
</head>
<body>


</body>
</html>
```

حال سایر دستورات این آموزش را در داخل تگ script وارد می نمائیم.

۳.تعریف آرایه در Java Script

یکی از روش های تعریف آرایه در Java Script این صورت :

```
var list = new Array();
```

۴.مقدار دادن به آرایه

می خواهیم به این آرایه ۳ مقدار اضافه کنیم ، برای مقدار دادن به خانه های یک آرایه می توانیم از حلقه for به این شکل استفاده کنیم :

```
for (var i = 0; i < 3; i++) {
   list[i] = i;
}
```

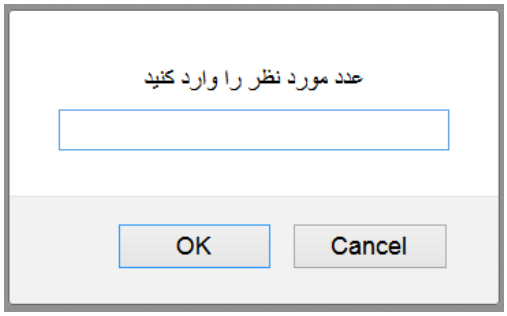
به این ترتیب مقادیر ۰ و ۱ و ۲ به ترتیب در خانه های ۰ و ۱ و ۲ قرار می گیرند.

می توانیم با تابع prompt مقداری را از ورودی دریافت کنیم ، با استفاده از قطعه کد زیر می توانیم تمامی خانه های آرایه را پُر کنیم :

```
for (var i = 0; i < 3; i++) {
   list[i] = parseInt(prompt("نمره را وارد کنید"));
}
```

تابع parseInt مقدار ورودی را به عدد تبدیل می کند. این کار برای مقایسه و مرتب سازی اعداد الزامی است ، در غیر این صورت چنانچه اعدادی با تعداد ارقام متفاوت وارد شوند عمل مقایسه به درستی انجام نمی شود.

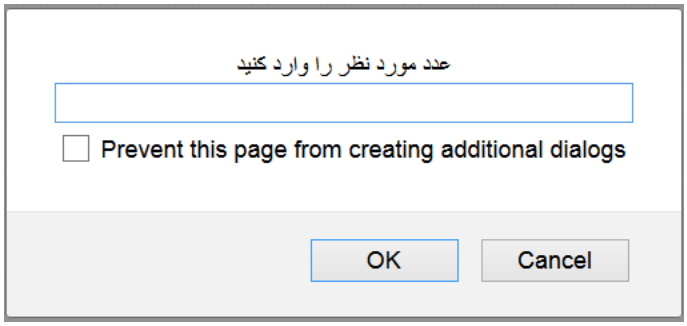
با اجرای این کد در مرورگر سه مرتبه (به تعداد دفعات تکرار حلقه for) این کادر نمایش داده می شود :



و از شما مقادیر را دریافت و در خانه های آرایه قرار می دهد.

نکته : در مرورگرهای مختلف ظاهر این پنجره متفاوت می باشد.

نکته : فقط در دفعه اول پنجره به این صورت نمایش داده می شود ، دفعات دوم به بعد کادر به شکل زیر تغییر پیدا می کند.



۵.چاپ آرایه

چاپ کردن آرایه را می توان همانند مقدار دادن به آرایه در نظر گرفت. با این تفاوت که در هنگام چاپ شدن بایستی تعیین کنیم که مستقیم چاپ شود و یا در تگ خاصی عمل چاپ انجام شود.

برای اینکه مستقیماً عمل چاپ را در صفحه انجام دهیم می توانیم از تابع write مربوط به آبجکت document به این صورت استفاده کنیم :

```
for (var i = 0; i < 3; i++) {
   document.write(list[i] + ‘<br/>’);
}
```

با استفاده از تگ
 هر مقدار در یک سطر چاپ می شود.

و برای اینکه عمل چاپ را در یک تگ Table انجام دهیم می توانیم به این صورت :

```
document.write("<table>");
for (var i = 0; i < 3; i++) {
   document.write("<tr>");
      document.write("<td>" + list[i] + "</td>");
   document.write("</tr>");
}
document.write("</table>");
```

به این ترتیب جدولی با سه ردیف که ردیف شامل یک ستون است ایجاد می شود و اطلاعات داخل این جدول نمایش داده می شود.

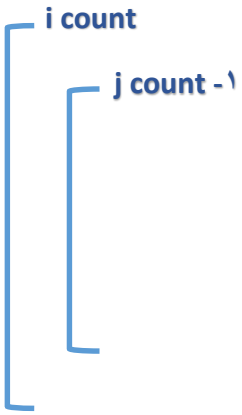
نکته : خروجی ظاهراً با حالت قبل یکسان است ، برای اینکه خطوط جدول را مشاهده کنید می توانید خصوصیت border مربوط به تگ table را مقدار دهی نمائید :

```
document.write("<table border='1'>");
```

۶.مرتب سازی آرایه ها در JavaScript

برای انجام مرتب سازی حبابی [۱] بایستی این دو عمل را انجام دهیم :

۱. دو حلقه for تو در تو ، حلقه j یکی کمتر از حلقه i.



علت کمتر بودن حلقه تکرار دوم ، مقایسه هر خانه آرایه با خانه ی بعدی خود است. سه خانه آرایه را به این صورت در نظر بگیرید :

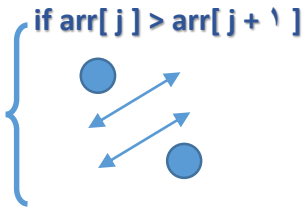
۵	۰
۹	۱
۶	۲

[۱] Bubble Sort

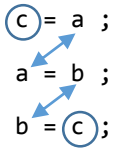
حال اگر در خانه دوم این آرایه باشیم (مقدار ۶) با کدام خانه بایستی مقایسه بایستی صورت بگیرد؟! چون خانه دیگری وجود ندارد خطا رخ می دهد و یا اینکه خروجی مورد نظر شما نمایش داده نمی شود.بنابراین چون می خواهیم هر خانه را با خانه ی بعدی آن مقایسه کنیم بایستی حلقه تکرار دوم را یک واحد کمتر از طول آرایه در نظر بگیریم.

نکته : برای مرتب سازی حبابی در JavaScript دو حلقه تکرار به یک اندازه باشند خطایی رخ نمی دهد اما در زبان C# با خطا (استثناء) IndexOutOfRangeException روبرو می شوید. در هر صورت در این آموزش حلقه تکرار دوم را یک واحد کمتر در نظر می گیریم.

۲. پس از بررسی یک خانه از آرایه با خانه بعدی ، در صورت بزرگ تر یا کوچک تر بودن (بسته به نوع مرتب سازی صعودی یا نزولی) عمل جابجایی مقدار دو خانه از آرایه انجام می گیرد. این عمل به ازای تک تک خانه های آرایه بایستی انجام شود تا در پایان لیست مرتب شود.



برای عمل جابجایی متغییرها a,b را در نظر بگیرید ، حال می خواهیم مقدار این دو متغییر را به کمک یک متغییر کمکی به نام c جابجا کنیم،در این حالت می توانیم به این صورت عمل کنیم:



در مقالات معمولا از این متغییر سوم که برای جابجایی استفاده می شود با نام temp یاد می شود به هر صورت نام گذاری متغییر از قوانین خود پیروی می کند و این نام دلخواه است همچنین تأثیری در روند اجرای برنامه ندارد.

```
temp = a;
a  = b;
b  = temp;
```

۷.کد برنامه

```
<script type="text/javascript">
   var count = 3;
   var list = new Array(count);
   for (var i = 0; i < count; i++) {
       list[i] = parseInt(prompt("نمره را وارد کنید"));
   }
   temp = 0;
   for (var i = 0; i < count; i++) {
       for (var j = 0; j < count - 1; j++) {
           if (list[j] < list[j + 1] ) {
               temp  = list[j];
               list[j] = list[j + 1];
               list[j + 1] = temp;
           }
       }
   }
</script>
```

تصویرسازی ذهنی برای الگوریتم ها، به کمک چند شکل ساده (دایره، براکت، آکولاد و...) در ابتدای برنامه بنویسی برای به نتیجه رسیدن ، به خاطر سپاری ، لذت بردن از برنامه نویسی بسیار تأثیر گذار و مؤثر است. این روش به شما کمک می کند تا با علاقه بیشتری برنامه نویسی را دنبال کنید. در ادامه و با تمرین ، تایپ دستورات ، نمونه کدها و مثال ها و مخصوصاً بیان مطالب برای دیگران (**آموزش**) می توانید برنامه نویسی را به خوبی یاد بگیرید.

۸.استفاده از توابع در برنامه

دستوراتی مثل تعریف آرایه ، مقدار دادن به آرایه و چاپ آن را فراگرفتید بهتر است برنامه را به قسمت های کوچک تری(توابعی)تقسیم بندی کنیم تاکنترل برنامه وخطایابی ساده‌تر شود. یکی از روش های ایجاد تابع در JavaScript به این صورت می باشد :

```
function نام_تابع() {
}
```

برای مثال تابع func۱ پیغامی را با متن JavaScript را نمایش می دهد :

```
function func1() {
   alert('JavaScript');
}
```

۹.تابع گرفتن مقادیر از ورودی (نام خانوادگی ، نمره)

ابتدا دو آرایه را در تگ JavaScript تعریف می کنیم سپس در توابع از این دو استفاده می کنیم :

```
<script type="text/javascript">
   var count = 3;
   var family = new Array(count);
   var score = new Array(count);
```

حال تابعی با نام inputData را برای دریافت نام خانوادگی و نمره از کاربر به این صورت تعریف می کنیم :

```
   function inputData() {
       for (var i = 0; i < count; i++) {
           family[i] = prompt("نام خانوادگی");
           score[i] = parseInt(prompt("نمره"));
       }
   }
```

۱۰.تابع چاپ مقادیر دریافت شده (در یک جدول)

سپس تابعی به نام print() برای چاپ مقادیر وارد شده در مرحله قبل ایجاد می کنیم ، این تابع اطلاعات را در یک جدول چاپ می کند :

```
   function print() {
       var s = "";
       s = "<table border='1'>";
       s += "<tr>";
       s += "<th>نام خانوادگی</th>";
       s += "<th>نمره</th>";
       s += "</tr>";
```

```
        for (var i = 0; i < count; i++) {
            s += "<tr>";
            s += "<td>" + family[i] + "</td><td>" + score[i] + "</td>";
            s += "</tr>";
        }
        s += "</table>";
        document.write(s);
    }
```

۱۱.تابع مرتب سازی براساس نام خانوادگی

این تابع به منظور مرتب سازی براساس نام خانوادگی ایجاد شده است.

```
    function sortByFamily() {
        temp = "";
        for (var i = 0; i < count; i++) {
            for (var j = 0; j < count - 1; j++) {
            if (family[j] > family[j + 1]) {
                temp = family[j];
                family[j] = family[j + 1];
                family[j + 1] = temp;

                temp = score[j];
                score[j] = score[j + 1];
                score[j + 1] = temp;
            }
        }
    }
}
```

نکته : به دلیل اینکه دو آرایه (نام خانوادگی و نمره) تعریف شده است در هنگام جابجایی بایستی در هر دو آرایه عمل جابجایی انجام شود.

۱۲.تابع مرتب سازی براساس نمره

و این تابع برای مرتب سازی براساس نمرات وارد شده تغییرات را اعمال می کند.

```
function sortByscore() {
    temp = "";
    for (var i = 0; i < count; i++) {
        for (var j = 0; j < count - 1; j++) {
            if (score[j] < score[j + 1]) {
                temp = family[j];
                family[j] = family[j + 1];
                family[j + 1] = temp;

                temp = score[j];
                score[j] = score[j + 1];
                score[j + 1] = temp;
            }
        }
    }
}
```

نکته : به دلیل اینکه دو آرایه (نام خانوادگی و نمره) تعریف شده است در هنگام جابجایی بایستی در هر دو آرایه عمل جابجایی انجام شود.

۱۳.یک مقداری زیباتر

می خواهیم تعدادی دکمه یا Button برروی صفحه داشته باشیم و سپس با کلیک بر روی هر کدام ازاین باتن ها عملی متناسب با عنوان آن انجام شود.

ابتدا در تگ body این چهار button را به همراه یک تگ به نام result برای نمایش خروجی لیست به این صورت اضافه می کنیم :

```
<body>
    <div>
        <input  id="btnAdd" value="لیست اسامی و نمرات" type="button"
onclick="btnAdd_click()" />
        <input id="btnView" value="نمایش لیست" type="button" onclick="btnView_click()"/>
        <input id="btnSortByScore" value="مرتب سازی براساس نمره" type="button"
onclick="btnSortByScore_click()" />
        <input id="btnSortByFamily" value="مرتب سازی براساس فامیل" type="button"
onclick="btnSortByFamily_click()" />
    </div>
    <div id="result">
    </div>
</body>
```

برای هر کدام از این باتن ها یک رویداد onClick تعریف شده است.عنوانی که برای رویدادها تعریف شده اند دلخواه می باشد اما بهتر است مشخص کننده نام کنترل و عنوان رویداد باشد.

برای مثال برای عمل افزودن عنوان باتن را btnAdd و عنوان رویدادکلیک این باتن را btnAdd_Click() قرار داده ایم تا در کد نویسی دسترسی بهتری داشته باشیم.

سپس در تگ JavaScript برای تمامی button های صفحه با همان عنوان یک تابع ایجاد می کنیم.

این تابع btnAdd_Click() در حقیقت رویدادی است که به یک باتن مربوط و با عمل کاربر – دراینجا کلیک شدن بر روی button – پاسخ می دهد و دستورات این بخش اجرا می شود.

```
function btnAdd_click() {
}
```

```
function btnView_click() {
}
function btnSortByScore_click() {
}
function btnSortByFamily_click() {
}
```

از توابع ایجاد شده در مراحل قبل می توانیم در این رویدادها استفاده کنیم ، به این ترتیب کد رویدادها به این صورت تغییر پیدا می کنند :

```
function btnAdd_click() {
   inputData();
}
function btnView_click() {
   print();
}
function btnSortByScore_click() {
   sortByscore();
   print();
}
function btnSortByFamily_click() {
   sortByFamily();
   print();
}
```

تابع print که در قبل ایجاد شده است اطلاعات را به این صورت چاپ می کند :

```
document.write(s);
```

این دستور باعث می شود صفحه به نوعی refresh شود ، به جای این دستور می توانیم یک div در داخل صفحه قرار دهیم ، عنوانی رابه آن اختصاص دهیم ، سپس با استفاده از تابع getElementById در Java Script این div را دریافت می کنیم و اطلاعات را به جای چاپ کردن در تمام صفحه تنها در همان div چاپ می کنیم که باعث زیباتر شدن کار می شود.

این div با عنوان result در قسمت باتن ها ایجاد شده است:

```
<div id="result">
</div>
```

حال به جای استفاده از :

```
document.write(s);
```

از :

```
var r = document.getElementById('result');
r.innerHTML = s;
```

استفاده می کنیم. در این قسمت متغییری به نام r تعریف شده است که با کمک تابع getElementById دنبال این Id در صفحه می گردد. سپس با استفاده از خصوصیت innerHTML ، تگ های ایجاد شده مربوط به جدول ، نام خانوادگی و نمرات را در این تگ چاپ می کنیم.

۱۴.جدول خلاصه دستورات

ردیف	عملیات	عنوان تابع	عنوان رویداد	توابع فراخوانی شده در رویداد
۱.	دریافت اطلاعات	inputData()	btnAdd_click()	inputData()
۲.	نمایش اطلاعات	print()	btnView_click()	print()
۳.	مرتب سازی نام	sortByFamily()	btnSortByFamily_click()	sortByFamily() print()
۴.	مرتب سازی نمره	sortByscore()	btnSortByScore_click()	sortByscore() print()

۱۵.کدهای برنامه

```
<head>
    <title></title>
    <script type="text/javascript">
        var count = 3;
        var family = new Array(count);
        var score = new Array(count);
        function inputData() {
            for (var i = 0; i < count; i++) {
                family[i] = prompt("نام خانوادگی");
                score[i] = parseInt(prompt("نمره"));
            }
        }
        function sortByFamily() {
            temp = "";
            for (var i = 0; i < count; i++) {
                for (var j = 0; j < count - 1; j++) {
                    if (family[j] > family[j + 1]) {
                        temp = family[j];
                        family[j] = family[j + 1];
                        family[j + 1] = temp;

                        temp = score[j];
                        score[j] = score[j + 1];
                        score[j + 1] = temp;
                    }
                }
            }
        }
        function sortByscore() {
            temp = "";
            for (var i = 0; i < count; i++) {
```

```
            for (var j = 0; j < count - 1; j++) {
                if (score[j] < score[j + 1]) {
                    temp = family[j];
                    family[j] = family[j + 1];
                    family[j + 1] = temp;

                    temp = score[j];
                    score[j] = score[j + 1];
                    score[j + 1] = temp;
                }
            }
        }
    }
    function print() {
        var s = "";
        s = "<table border='1'>";
        s += "<tr>";
        s += "<th>نام خانوادگی</th>";
        s += "<th>نمره</th>";
        s += "</tr>";
        for (var i = 0; i < count; i++) {
            s += "<tr>";
            s += "<td>" + family[i] + "</td><td>" + score[i] + "</td>";
            s += "</tr>";
        }
        s += "</table>";
        var r = document.getElementById('result');
        r.innerHTML = s;
    }
    function btnAdd_click() {
        inputData();
    }
```

```
        function btnView_click() {
            print();
        }
        function btnSortByScore_click() {
            sortByscore();
            print();
        }
        function btnSortByFamily_click() {
            sortByFamily();
            print();
        }
    </script>
 </head>
 <body>
    <div>
        <input id="btnAdd" value="لیست اسامی و نمرات" type="button" onclick="btnAdd_click()"
/>
        <input id="btnView" value="نمایش لیست" type="button" onclick="btnView_click()" />
        <input id="btnSortByScore" value="مرتب سازی براساس نمره " type="button"
onclick="btnSortByScore_click()" />
        <input id="btnSortByFamily" value="مرتب سازی براساس نام " type="button"
onclick="btnSortByFamily_click()" />
    </div>
    <div id="result">
    </div>
 </body>
 </html>
```

پیروز و سربلند باشید.

محمدحسین عبدالهی

۱۳۹۴/۱/۱۹

Email : MH_SQL@yahoo.com
Web Site: www.FullCode.ir