

بسم الله الرحمن الرحيم

آموزشگاه تحلیل داده

تخصصی ترین مرکز برنامه نویسی و دیتابیس در ایران

توابع مرتبط با آرایه ها در جاوا اسکریپت

مدرس: مهندس افشین رفوآ

توابع مرتبط با آرایه ها در جاوا اسکریپت / JavaScript array methods

قدرت زبان برنامه نویسی تحت وب جاوا اسکریپت مبتنی بر توابع و متدهای توانمندی است که در آن بکار می رود.

تبدیل آرایه به رشته / array to string conversion

کلیه ی اشیا در زبان javascript دارای متدهای `valueOf()` و `toString()` می باشند.

تابع `valueOf()` رفتار یا عملکرد پیش فرض یک آرایه است. متد بیان شده یک آرایه را به صورت رشته (`string`) بازگردانی می نماید.

مثال

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];
document.getElementById("demo").innerHTML =
fruits.valueOf();
```

در زبان جاوا اسکریپت، هر دو متدهای `valueOf()` و `toString()` یکسان محسوب می شوند.

مثال

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];
document.getElementById("demo").innerHTML =
fruits.toString();
```

متد `join()` تمامی المان های آرایه را در قالب یک رشته به هم متصل می کند. عملکرد این متد درست مشابه `toString()` می باشد با این تفاوت که با استفاده از این تابع می توان `separator` (جدا کننده) بین المان های آرایه را نیز تعریف و مشخص کرد.

مثال

```
<p id="demo"></p>
<script>
var fruits = ["Banana", "Orange", "Apple", "Mango"];
document.getElementById("demo").innerHTML = fruits.join(" * ");
</script>
```

استفاده از توابع `push()` و `pop()`

به هنگام کار با آرایه ها، به راحتی می توان المان جدید به آن ها ضمیمه کرد و یا یک المان موجود را از آن حذف کرد.

توابع یا متدهای `pop ()` و `push ()` هم دقیقا همین کار را انجام می دهند.

تابع `pop ()` یک آیتم موجود را از انتهای آرایه ی مورد نظر حذف می کند.

مثال

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];
    fruits.pop(); // Removes the last element
("Mango") from fruits
```

این در حالی است که متد `push()` یک المان جدید به انتهای آرایه ی موجود اضافه می کند.

نکته

توجه داشته باشید که `[0]` اولین المان در یک آرایه می باشد و `[1]` دومین و همین طور ادامه می یابد.

اندیس (شمارنده) های آرایه از عدد `0` آغاز می گردند.

مثال

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];
    fruits.push("Kiwi"); // Adds a new element
("Kiwi") to fruits
```

متد `pop ()` رشته ای را برمی گرداند که از آرایه حذف گردیده، متد `push ()` تعداد المان های آرایه ی جدید را بازیابی می کند.

حذف اولین المان یک آرایه – تابع `shift`

Shifting یا جابجایی درست همانند متد `popping` (حذف المان از انتهای آرایه) عمل می کند با این تفاوت که

بجای المان پایانی، المان آغازین یک آرایه را دستکاری می کند.

تابع `shift()` اولین المان موجود در یک آرایه را حذف می کند.

مثال

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];
    fruits.shift();           // Removes the first element
"Banana" from fruits
```

در مقابل تابع **unshift** را داریم. این تابع یک المان به ابتدای آرایه ی جاری اضافه می کند.

مثال

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];
    fruits.unshift("Lemon"); // Adds a new element
"Lemon" to fruits
```

تابع **shift ()** رشته ای را باز می گرداند که از ابتدای آرایه حذف گردیده است.

متد **unshift ()** تعداد المان های جدید آرایه را بازمی گرداند.

هر المان در آرایه دارای یک شماره ی (اندیس) منحصر بفرد است که می توان از طریق آن به المان مورد نظر دسترسی پیدا کرد.

مثال

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];
    fruits[0] = "Kiwi";       // Changes the first element
of fruits to "Kiwi"
```

با استفاده از خاصیت **length (property)** می توان به یک آرایه، المان جدید افزود.

مثال

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];
    fruits[fruits.length] = "Kiwi"; // Appends
"Kiwi" to fruit
```

از آنجایی که آرایه ها در جاوا اسکریپت خود شی هستند، می توان المان ها را با استفاده از عملگر **delete** حذف کرد.

مثال

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];
    delete fruits[0];           // Changes the first element
in fruits to undefined
```

تابع splice

به وسیله ی این متد می توان آیتم های جدید به یک آرایه افزود. اضافه کردن یک عنصر جدید در مکان (قرار گیری) 2 یک آرایه از طریق این متد صورت می گیرد.

مثال

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];
    fruits.splice(2, 0, "Lemon", "Kiwi");
```

اولین پارامتر (عدد 2) موقعیتی که المان جدید در آن به آرایه ی مورد نظر اضافه می شود را تعیین می کند. دومین پارامتر (0) تعداد المان هایی که باید حذف شوند را مشخص می کند.

باقی پارامترها ("Lemon" و "Kiwi") نیز المان های جدیدی که قرار است به آرایه اضافه شوند را تعریف می کنند.

حذف المان ها با استفاده از آرایه با استفاده از splice()

با استفاده از متد splice () همچنین می توان المان جدید به یک آرایه اضافه کرد.

مثال

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];
fruits.splice(0, 1); // Removes the first element of fruits
```

اولین پارامتر (0) گوینا و تعیین کننده ی مکانی است که المان جدید در آن به آرایه اضافه می شود.

دومین پارامتر (1) تعداد المان هایی که باید از آرایه ی جاری حذف گردند را مشخص می کند.

بقیه ی پارامترها حذف گشته و هیچ المان جدیدی اضافه نمی گردد.

مرتب سازی آرایه به ترتیب حروف الفبا

به کمک تابع `sort()` می توان یک آرایه را به ترتیب حروف الفبا مرتب کرد.

مثال

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];
    fruits.sort(); // Sorts the elements of
fruits
```

وارونه کردن ترتیب عناصر آرایه

با استفاده از تابع `reverse()` می توان ترتیب (قرارگیری) المان های موجود در یک آرایه را معکوس کرد.

مثال

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];
    fruits.sort(); // Sorts the elements of
fruits
    fruits.reverse(); // Reverses the order of the
elements
```

مرتب سازی عددی

به صورت پیش فرض تابع `sort () (function)` مقادیر را به صورت رشته یا به عنوان گروهی از رشته ها مرتب می کند. این امر در مرتب سازی رشته ها بسیار کارآمد عمل می کند ("`Apple`" پیش از "`Banana`" قرار می گیرد). استفاده از این تابع در حالت پیش فرض آن برای مرتب سازی اعداد نتایج ناصحیح بدست می دهد. به عنوان مثال "`25`" از "`100`" بزرگتر است زیرا که "`2`" از "`1`" بزرگتر محسوب می شود. به این خاطر است که متد مزبور در مرتب سازی اعداد منجر به تولید نتایج ناصحیح می شود.

جهت رفع این مشکل می توان از یک `compare function` (تابع مقایسه ای) بهره گرفت. در مثال زیر آرایه به ترتیب صعودی مرتب می شود.

مثال

```
var points = [40, 100, 1, 5, 25, 10];
    points.sort(function(a, b){return a-b});
```

با استفاده از روش بالا می توان آرایه ی بیان شده را به ترتیب نزولی نیز مرتب سازی کرد.

مثال

```
var points = [40, 100, 1, 5, 25, 10];
points.sort(function(a, b){return b-a});
```

تابع مقایسه ای (compare function)

به کمک تابع مقایسه ای می توان یک روش مرتب سازی جایگزین برای المان های آرایه تعریف کرد.

این تابع باید با توجه به آرگومان ها یک مقدار منفی، صفر یا مقدار مثبت بازگردانی نماید.

```
function(a, b){return a-b}
```

تابع `sort()` دو مقدار را با هم مقایسه کرده و آن ها را به تابع مقایسه ای `(compare function)` ارسال می کند. سپس مقادیر مذکور را با توجه به مقدار بازگشتی `(return value)` مرتب می سازد.

مثال

زمانی که مقادیر 40 و 100 را با هم مقایسه می کنیم، متد `sort()` تابع مقایسه ای `(40,100)` را فرا می خواند.

تابع مزبور عملیات 40-100 را محاسبه کرده و مقدار -60 که مقداری منفی می باشد را باز می گرداند.

پیدا کردن بیشترین (یا کمترین) مقدار یک آرایه

چگونه می توان بیشترین مقدار را در یک آرایه یافت؟ در مثال زیر مقادیر آرایه به ترتیب نزولی مرتب سازی می شوند.

مثال

```
var points = [40, 100, 1, 5, 25, 10];
points.sort(function(a, b){return b-a});
// now points[0] contains the highest value
```

روش پیدا کردن کمترین مقدار نیز در مثال زیر تشریح شده است. در مثال زیر مقادیر به ترتیب صعودی مرتب سازی می شوند.

مثال

```
var points = [40, 100, 1, 5, 25, 10];
points.sort(function(a, b){return a-b});
// now points[0] contains the lowest value
```

توابع متصل سازی آرایه ها

متد `concat()` با اتصال دو آرایه به هم، آرایه ای جدید خلق می کند.

مثال

```
var myGirls = ["Cecilie", "Lone"];
var myBoys = ["Emil", "Tobias", "Linus"];
var myChildren = myGirls.concat(myBoys); //
Concatenates (joins) myGirls and myBoys
```

تابع یاد شده قادر است هر تعداد آرگومان آرایه که لازم باشد، بگیرد.

مثال

```
var arr1 = ["Cecilie", "Lone"];
var arr2 = ["Emil", "Tobias", "Linus"];
var arr3 = ["Robin", "Morgan"];
var myChildren = arr1.concat(arr2, arr3); //
Concatenates arr1 with arr2 and arr3
```

تابع `slice()`

متد `slice()` یک تکه از آرایه ی موجود را برش داده و داخل آرایه ی دیگری می ریزد.

مثال

```
var fruits = ["Banana", "Orange", "Lemon", "Apple", "Mango"];
var citrus = fruits.slice(1, 3)
```