

# بسم الله الرحمن الرحيم

## آموزشگاه تحلیل داده

تخصصی ترین مرکز برنامه نویسی و دیتابیس در ایران

آموزش آرایه ها در javascript

مدرس: مهندس افشین رفوآ

## آموزش آرایه ها در javascript

آرایه ها (در جاوا اسکریپت) چندین متغیر را در قالب تنها یک متغیر ذخیره می کنند یا به عبارتی دیگر آرایه یک متغیر خاص است که می توانید به طور همزمان چندین مقدار (بیش از یک مقدار) را در آن ذخیره نمایید. در این بخش با استفاده از یک اسکریپت، آرایه ها را داخل تگ `<p>` با شناسه `(id) "demo"` نمایش می دهیم.

### مثال

```
<p id="demo"></p>
<script>
var cars = ["Saab", "Volvo", "BMW"];
document.getElementById("demo").innerHTML = cars;
</script>
```

همان طور که در مثال بالا مشاهده می کنید، اولین خط آرایه ای به نام cars ایجاد می کند.

دومین خط المانی که شناسه ی آن `"demo"` می باشد را یافته سپس آرایه را در `"innerHTML"` آن المان نمایش می دهد.

### مثال

```
<p id="demo"></p>
<script>
var cars = ["Saab", "Volvo", "BMW"];
document.getElementById("demo").innerHTML = cars;
</script>
```

خودتان امتحان کنید.

یک آرایه ایجاد کرده و مقادیری به آن تخصیص دهید .

```
var cars = ["Saab", "Volvo", "BMW"];
```

فاصله و انتهای خط (`line break`) از هیچ اهمیتی برخوردار نیستند. یک تعریف (`declaration`) می تواند در چندین خط ادامه داشته باشد.

### مثال

```
var cars = [
    "Saab",
    "Volvo",
    "BMW"
];
```

**توجه:** توجه داشته باشید که هیچ گاه پس از آخرین المان (در مثال بالا "BMW") ویرگول قرار ندهید، زیرا ممکن است در مرورگرهای مختلف نتایج متفاوتی ارائه دهد.

### آرایه چیست؟

شی آرایه یک نوع خاص از متغیر هاست که می تواند چندین داده را در قالب یک نام در خود ذخیره کند. آرایه مجموعه ای از متغیرهایی از یک نوع داده ای با نام یکسان است، که هر کدام از اعضای آن توسط یک شماره یا شاخص (**index**)، از یکدیگر تمیز داده می شوند. به عنوان مثال فرض کنید فهرستی از ارقام یا آیتم ها (برای مثال لیستی از اسم ماشین ها) را دارید که هر یک در متغیر ی مجزا ذخیره می گردد.

```
var car1 = "Saab";
var car2 = "Volvo";
var car3 = "BMW";
```

چنانچه بخواهیم درون آیتم های فوق حلقه زده (آن ها را تکرار کرده) و ماشین مورد نظر را پیدا کنیم چه اتفاقی رخ می دهد؟ اگر تعداد ماشین ها 3 تا نه بلکه 300 تا بود چه اتفاقی رخ می داد؟ این مسئله با بهره گیری از آرایه حل می شود.

یک آرایه همان طور که پیش تر ذکر کردیم قادر است چندین مقدار را تحت نام یکسان در خود ذخیره کرده و از طریق اندیس به آن مقادیر دسترسی پیدا کند.

### ایجاد آرایه

استفاده از یک آرایه ی متنی (**array literal**) بهترین و نیز آسان ترین روش برای ایجاد یک آرایه ی جاوا اسکریپت می باشد.

## دستور نگارش (syntax)

```
var array-name = [item1, item2, ...];
```

مثال

```
var cars = ["Saab", "Volvo", "BMW"];
```

## استفاده از کلید واژه `new`

مثال زیر نیز یک آرایه ایجاد کرده و به آن مقادیری را تخصیص می دهد.

مثال

```
var cars = new Array("Saab", "Volvo", "BMW");
```

نکته

دو مثال بالا هر دو یک کار یکسان را انجام می دهند. نیازی به استفاده از `new Array()` وجود ندارد. برای آسانی، قابلیت خوانایی بیشتر و نیز افزایش سرعت اجرا توصیه می کنیم از نمونه ی اول استفاده کنید.

## دسترسی به المان های آرایه

برای دسترسی به المان مورد نظر در آرایه از شماره ی اندیس کمک می گیریم.

دستور زیر نحوه ی دسترسی به مقدار اولین المان را نمایش می هد .

```
var name = cars[0];
```

دستور زیر اولین المان را اصلاح می کند .

```
cars[0] = "Opel";
```

توجه

دقت داشته باشید که `[0]` اولین المان محسوب می گردد و `[1]` دومین شمارنده ی آرایه ها در جاوا اسکریپت از صفر آغاز می شود.

## امکان داشتن چندین شی مختلف در یک آرایه

متغیر در جاوا اسکریپت می تواند یک شی باشد. در حقیقت آرایه یک نوع خاص شی محسوب می گردد، به این خاطر هم می توانید متغیرهایی از نوع های مختلف در یک آرایه داشته باشیم.

می توان اشیا (object)، توابع (function) یا آرایه (array) در دل آرایه ی دیگر قرار داد.

```
myArray[0] = Date.now;
myArray[1] = myFunction;
myArray[2] = myCars;
```

### آرایه یک شی است

آرایه یک نوع خاص شی است. عملگر **typeof (operator)** در جاوا اسکریپت (برای آرایه) **"object"** باز می گرداند.

آرایه برای دسترسی به المان های خود از شماره کمک می گیرد. به عنوان مثال **person[0]** المان **John** را باز می گرداند.

### مثال Array

```
var person = ["John", "Doe", 46];
```

اشیا با استفاده از اسم (**name**) به اعضای ("**member**") خود دستیابی پیدا می کنند. برای مثال **person.firstName** المان **John** را بازیابی می کند.

### مثال Object

```
var person = {firstName:"John", lastName:"Doe", age:46};
```

### property ها (خواص) و method های (توابع) مربوط به آرایه

اگر به مثال زیر دقت کنید متوجه می شوید که **length property** تعداد المان های موجود در **cars** را بر می گرداند و تابع **cars.sort()** را به ترتیب حروف الفبا مرتب سازی (**sort**) می کند.

### مثال

```
var x = cars.length;           // The length property returns the
number of elements in cars
    var y = cars.sort();       // The sort() method sort
cars in alphabetical order
```

## تشریح خاصیت length

**length property** یک آرایه، طول یا تعداد المان های موجود در یک آرایه را باز گردانی می کند.

### مثال

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];
    fruits.length;           // the length of
fruits is 4
```

## افزودن المان جدید به یک آرایه

آسان ترین روش برای افزودن یک **element** جدید به آرایه، استفاده از تابع **push (method)** می باشد.

### مثال

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

همچنین می توان این کار را با بهره گیری از **length property** انجام داد.

### مثال

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

## ایجاد یک حلقه در آرایه (تکرار المان های یک آرایه به تعداد دفعات معین)

در **for loop** مجموعه دستورالعمل ها به تعداد معلوم و مورد نیاز، تکرار می شوند. یکی از بهترین روش های

تکرار المان های یک آرایه استفاده از حلقه ی بیان شده است.

### مثال

```

var index;
var fruits = ["Banana", "Orange", "Apple", "Mango"];
for (index = 0; index < fruits.length; index++) {
    text += fruits[index];
}

```

آرایه ی شرکت پذیر چیست (associative arrays)؟

بسیاری از زبان های برنامه نویسی کنونی از آرایه های با اندیس های نام گذاری شده پشتیبانی می کنند.

آرایه هایی که اندیس های آن ها نام گذاری شده، آرایه های شرکت پذیر یا **arrays associative (hashes)** نامیده می شوند.

جاوا اسکریپت از آرایه هایی که دارای **named index** (اندیس نام گذاری شده) هستند پشتیبانی نمی کند.

در عوض این زبان از قابلیت **numbered indexes** (اندیس های شماره گذاری شده) پشتیبانی می کند.

مثال

```

var person = [];
    person[0] = "John";
    person[1] = "Doe";
    person[2] = 46;
    var x = person.length;           // person.length will
return 3
    var y = person[0];              // person[0] will return
"John"

```

توجه

در صورت استفاده از آرایه ی نام گذاری شده، جاوا اسکریپت آرایه را به یک **object** استاندارد باز تعریف می کند. پس از آن تمامی متدها و خواص مربوط به آرایه نتایج نا صحیح و غیر مجاز تولید می کنند.

مثال

```

var person = [];
    person["firstName"] = "John";
    person["lastName"] = "Doe";
    person["age"] = 46;
    var x = person.length;           // person.length will
return 0

```

```
var y = person[0]; // person[0] will return
undefined
```

## تشریح تفاوت بین آرایه (array) و شی (object)

همان طور که پیش تر شرح داده شد، در جاوا اسکریپت آرایه ها از اندیس های شماره دار استفاده می کنند.

اشیا (object) نیز از اندیس های نام دار (named indexes) استفاده می کنند.

نکته: آرایه یک نوع خاصی شی با اندیس شماره دار می باشد.

## چه زمانی باید از array استفاده کرد و چه زمانی از object؟

JavaScript از آرایه های شرکت پذیر (associative array) پشتیبانی نمی کند.

اگر می خواهید اسم المان ها از نوع رشته (نوشته یا متن) باشد، در آن صورت به طور حتم باید از

object استفاده کنید.

چنانچه مایلید اسم المان از نوع عددی یا شماره باشد، باید از array استفاده کنید.

تا حد ممکن سعی کنید از بکار گیری constructor (سازنده) پیش فرض و توکار JavaScript، new Array() خودداری کنید.

بجای استفاده از روش منسوخ بالا می توانید از روش نوین [] بهره بگیرید.

هر دو دستور کاملا متفاوت نام برده، آرایه ای تهی به نام points ایجاد می کنند.

```
var points = new Array(); // Bad
var points = []; // Good
```

بار دیگر هر دو دستور آرایه ای ایجاد می کنند که دربردارنده ی 6 عدد است.

```
var points = new Array(40, 100, 1, 5, 25, 10) // Bad
var points = [40, 100, 1, 5, 25, 10]; // Good
```

استفاده از کلیدواژه ی new برای ایجاد آرایه ی جدید به دلیل طولانی و پیچیده تر کردن کد منسوخ تلقی

تلقی می گردد.



```
var points = new Array(40, 100); // Creates an array with two
elements (40 and 100)
```

ببینید در صورت حذف کردن یکی از المان ها چه اتفاقی رخ می دهد!

```
var points = new Array(40); // Creates an array with 40
undefined elements !!!!!
```

### چگونه یک آرایه را تشخیص دهیم؟

یکی از سوال هایی که به طور متدوال مطرح می شود این است که چگونه می توان تشخیص داد آیا یک متغیر آرایه هست یا خیر؟

یکی از مشکل هایی که با آن مواجه می شویم، این است که عملگر `typeof` "object" باز می گرداند.

```
var fruits = ["Banana", "Orange", "Apple", "Mango"];
typeof fruits; // typeof returns object
```

عملگر مزبور به این خاطر "object" برمی گرداند که یک `JavaScript array` یا آرایه ی جاوا اسکریپت خود یک شی است.

جهت رفع این مشکل توصیه می کنیم `isArray() function` خود را ایجاد کنید.

```
function isArray(myArray) {
    return
myArray.constructor.toString().indexOf("Array") > -1;
}
```

`function` (تابع) بالا همیشه در صورتی که آرگومان مورد نظر یک آرایه باشد، مقدار `true` باز می گرداند و یا اگر بخواهیم دقیق تر توضیح بدیم اگر نوع ورودی تابع از نوع "[object array]" باشد مقدار صحیح (`true`) برمی گرداند.