بسم الله الرحمن الرحیم

آموزشگاه تحلیل داده

تخصصی ترین مرکز برنامه نویسی و دیتابیس در ایران

نمونه های اولیه ی شی (object prototype)

مدرس : مهندس افشین رفوآ

نمونه های اولیه ی شی (object prototype)

هر شی ای که در جاوا اسکریپت با آن سر و کار داریم، یک نمونه ی اولیه (prototype) دارد.

Prototype، یک خاصیت متعلق به تمامی اشیا جاوا اسکریپت است که اشیا قادراند از طریق آن قابلیت های جدیدی به ارث ببرند.

در واقع کلیه ی اشیا جاوا اسکریپت، خاصیت ها و متدهای خود را از نمونه ی اولیه خود به ارث می برند.

Prototype های جاوا اسکریپت

همان طور که پیش تر شرح دادیم، تمامی اشیا جاوا اسکریپت خاصیت ها و توابع خود را از نمونه ی اولیه ی خود به ارث می گیرند.

اشیایی که با استفاده از object literal ها و یا توسط تابع new Object() ایجاد می شوند، در واقع دارند از یک نمونه ی اولیه به نام Object.prototype خاصیت ها و توابع خود را به ارث می گیرند.

اشیایی که به وسیله ی تابع new Date() ایجاد شده اند، Date.prototype را به ارث می برند.

Object.prototype در بالاترین مرتبه زنجیره ی متشکل از نمونه های اولیه قرار می گیرد.

تمامی اشیا جاوا اسکریپت (اعم از Date، Array، RegExp، Function و غیره ... .) خواص و متدهای خود را از Object.prototype به ارث می برند.

نحوه ی ایجاد نمونه ی اولیه

شیوه ی متداول و استاندارد ایجاد نمونه ی اولیه شی (object prototype)، استفاده از یک constructor object function (استفاده از یک تابع به عنوان سازنده) است.

مثال:

```
function person(first, last, age, eyecolor) {
this.firstName = first;
```

```
this.lastName = last;
this.age = age;
this.eyecolor = eyecolor;
}
```

در این روش می توان به کمک کلیدواژه ی new، از نمونه ی اولیه یکسان اشیا (چندین شی متعدد) جدید ایجاد کرد:

```
<!DOCTYPE html>
<html>
<body>
<p id="demo"></p>
<script>
function person(first, last, age, eyecolor) {
this.firstName = first;
this.lastName = last;
this.age = age;
this.eyecolor = eyecolor;
}
var myFather = new person("John", "Doe", 50, "blue");
var myMother = new person("Sally", "Rally", 48, "green");
document.getElementById("demo").innerHTML =
"My father is " + myFather.age + ". My mother is " + myMother.age;
</script>
</body>
</html>
```

constructor function، نمونه ی اولیه برای شی person است.

افزودن خواص و توابع جدید به اشیا

گاهی شما نیاز دارید property (خواص) یا method های (توابع) جدیدی به شی موجود یا اشیا موجود از نوع خاص و یا نمونه ی اولیه ی یک شی اضافه کنید. نحوه ی افزودن خاصیت جدید به تمامی موارد نام برده ر ا زیر برای شما شرح می دهیم:

افزودن خاصیت جدید به شی موجود:

اضافه کردن property جدید به شی موجود بسیار آسان می باشد. برای یادگیری روش آن کافی است به مثال زیر دقت کنید:

```
<!DOCTYPE html>
<html>
<body>
  <p id="demo"></p>
  <script>
    function person(first, last, age, eyecolor) {
      this.firstName = first;
      this.lastName = last;
      this.age = age;
```

```
    this.eyecolor = eyecolor;
  }
  var myFather = new person("John", "Doe", 50, "blue");
  var myMother = new person("Sally", "Rally", 48, "green");
  myFather.nationality = "English";
  document.getElementById("demo").innerHTML =
  "My father is " + myFather.nationality;
</script>
</body>
</html>
```

همان طور که مشاهده می کنید، خاصیت جدید به شی myFather اضافه می شود (و نه به شی myMother و یا هر شی دیگر).

افزودن متد جدید به یک شی موجود:

```
<!DOCTYPE html>
<html>
<body>
  <p id="demo"></p>
  <script>
    function person(first, last, age, eyecolor) {
      this.firstName = first;
      this.lastName = last;
      this.age = age;
      this.eyecolor = eyecolor;
    }
    var myFather = new person("John", "Doe", 50, "blue");
    var myMother = new person("Sally", "Rally", 48, "green");
    myFather.name = function () {
      return this.firstName + " " + this.lastName;
    };
    document.getElementById("demo").innerHTML =
    "My father is " + myFather.name();
  </script>
</body>
</html>
```

متد مورد نظر به شی myFather اضافه می شود و نه به شی myMother.

افزودن خاصیت های جدید به یک نمونه ی اولیه:

نمی توان به همان شیوه ای که یک property جدید را به شی موجود ضمیمه می کردیم، یک خاصیت جدید به نمونه ی اولیه (prototype) اضافه کنید زیرا که نمونه ی اولیه یک شی موجود محسوب نمی شود.

مثال:

به منظور افزودن خاصیت جدید به یک سازنده (constructor)، باید آن را به تابع سازنده (تابعی که به عنوان سازنده استفاده می شود) اضافه کنید:

```
function person(first, last, age, eyecolor) {
    this.firstName = first;
```

```
    this.lastName = last;
    this.age = age;
    this.eyecolor = eyecolor;
    this.nationality = "English"
}
```

property های نمونه ی اولیه، می توانند مقدار داشته باشند (مقادیر پیش فرض).

بکار بردن خاصیت prototype

خاصیت prototype به شما این امکان را می دهد که خواص جدید به یک نمونه ی اولیه ی موجود اضافه کنید:

```
function person(first, last, age, eyecolor) {
    this.firstName = first;
    this.lastName = last;
    this.age = age;
    this.eyecolor = eyecolor;
}
person.prototype.nationality = "English";
```

خاصیت مذکور همچنین به شما اجازه می دهد، متدهای جدید به نمونه ی اولیه ی موجود اضافه کنید:

```
function person(first, last, age, eyecolor) {
    this.firstName = first;
    this.lastName = last;
    this.age = age;
    this.eyecolor = eyecolor;
}
person.prototype.name = function() {
    return this.firstName + " " + this.lastName;
};
```

توجه: هیچگاه نمونه های اولیه ی اشیا استاندارد جاوا اسکریپت را اصلاح یا modify نکنید. در صورت نیاز سعی کنید، فقط و فقط نمونه های اولیه ای که خود ایجاد کرده اید را دستکاری کنید.