

CoffeeScript #5	عنوان:
وحید محمدطاهری	نویسنده:
۱۶:۵۵ ۱۳۹۴/۰۳/۳۱	تاریخ:
www.dotnettips.info	آدرس:
JavaScript, CoffeeScript	گروه‌ها:

Classes

کلاس نه تنها در جاوااسکریپت بلکه در سایر زبان‌ها از جمله CoffeeScript نیز، بسیار مفید است.

در پشت صحنه، CoffeeScript برای ایجاد کلاس از prototype استفاده می‌کند. برای ساختن کلاس در CoffeeScript از کلمه کلیدی `class` باید استفاده کنید.

```
class Animal
```

نتیجه‌ی کامپایل مثال بالا می‌شود:

```
var Animal;
Animal = (function() {
  function Animal() {}
  return Animal;
})();
```

در مثال بالا، `Animal` نام کلاس و همچنین نامی است که برای ایجاد یک نمونه از آن می‌توانید استفاده کنید. در پشت صحنه CoffeeScript با استفاده از سازنده توابع این کار را انجام می‌دهد. یعنی شما می‌توانید با استفاده از کلمه کلیدی `new` یک نمونه از کلاس نوشته شده را بسازید.

```
animal = new Animal
```

تعریف سازنده برای کلاس بسیار ساده است. فقط کافی است از کلمه کلیدی `constructor` به عنوان یک تابع در کلاس تعریف شده استفاده کنید. این تابع شبیه به `initialize` در Ruby و `__init__` در Python است.

```
class Animal
  constructor: (name) ->
    @name = name
```

نتیجه کامپایل کد بالا می‌شود:

```
var Animal;
Animal = (function() {
  function Animal(name) {
    this.name = name;
  }
  return Animal;
})();
```

همچنین CoffeeScript امکان خلاصه نویسی را در سازنده کلاس نیز ایجاد کرده است. برای اینکار با اضافه کردن `@` به آرگومان‌های تابع سازنده به صورت پیشنهاد این کار انجام می‌شود. مثال زیر معادل مثال قبل است که به صورت دستی مقدار دهی انجام شده است.

```
class Animal
  constructor: (@name) ->
```

و برای استفاده از این کلاس

```
animal = new Animal "Cat"
alert "Animal is a #{animal.name}"
```

Instance properties اضافه کردن property به یک کلاس بسیار ساده و راحت است، syntax این کار دقیقا مانند اضافه کردن property به یک object است. فقط نکته ای که باید رعایت شود میزان تو رفتگی property نوشته شده است که به طور صحیح در داخل بدنه کلاس قرار بگیرد.

```
class Animal
  price: 5

  sell: (customer) ->

animal = new Animal
animal.sell(new Customer)
```

نتیجه کامپایل کد بالا می شود:

```
var Animal, animal;

Animal = (function() {
  function Animal() {}

  Animal.prototype.price = 5;

  Animal.prototype.sell = function(customer) {};

  return Animal;
})();

animal = new Animal;
animal.sell(new Customer);
```

Static properties

برای تعریف property به صورت static باید کلمه کلیدی this را به ابتدای آن اضافه کنید.

```
class Animal
  this.find = (name) ->

Animal.find("Dog")
```

در قسمت های قبل گفته شد، که به جای this می توان از @ استفاده کرد، در اینجا نیز می توان چنین کاری را انجام داد.

```
class Animal
  @find: (name) ->

Animal.find("Dog")
```

نتیجه ی کامپایل آن می شود:

```
var Animal;

Animal = (function() {
  function Animal() {}

  Animal.find = function(name) {};

  return Animal;
})();
```

```
Animal.find("Dog");
```