

CoffeeScript #9	عنوان:
وحید محمدطاهری	نویسنده:
۱۹:۵۵ ۱۳۹۴/۰۴/۲۲	تاریخ:
<a href="http://www.dotnettips.info">www.dotnettips.info</a>	آدرس:
JavaScript, CoffeeScript	گروه‌ها:

## اصطلاحات عمومی CoffeeScript

### Multiple arguments

همانطوری که [در قسمت قبل](#) در تابع `Math.max` مشاهده کردید، با استفاده از ... آرایه را به عنوان آرگومان چندگانه به تابع `max` ارسال کردیم. در پشت صحنه CoffeeScript برای اطمینان از ارسال کامل آرایه به تابع `max`، برای فراخوانی از تابع `apply()` استفاده می‌کند. ما نیز می‌توانیم از این ویژگی در جای دیگری استفاده کنیم.

```
Log =
  log: ->
    console?.log(arguments...)
```

نتیجه‌ی کامپایل آن می‌شود:

```
var Log;

Log = {
  log: function() {
    return typeof console !== "undefined" && console !== null ? console.log.apply(console, arguments) :
    void 0;
  }
};
```

و یا می‌توان قبل از ارسال آرگومان‌ها در آنها تغییر ایجاد کرد.

```
Log =
  logPrefix: "(App)"

  log: (args...) ->
    args.unshift(@logPrefix) if @logPrefix
    console?.log(args...)
```

نتیجه‌ی کامپایل آن می‌شود:

```
var Log,
    slice = [].slice;

Log = {
  logPrefix: "(App)",
  log: function() {
    var args;
    args = 1 <= arguments.length ? slice.call(arguments, 0) : [];
    if (this.logPrefix) {
      args.unshift(this.logPrefix);
    }
    return typeof console !== "undefined" && console !== null ? console.log.apply(console, args) : void
    0;
  }
};
```

همانطور که مشاهده می‌کنید آرگومان‌های ارسالی به تابع `log` پس از چک شدن متغیر `logPrefix` و در صورت داشتن مقدار توسط تابع `unshift` به ابتدای آنها اضافه می‌شود.

### And/Or

طبق ساختار `syntax` ایی که در قسمت‌های قبل با آن آشنا شدیم، `or` به جای `||` و `and` به جای `&&` استفاده شده و سبب خوانایی

بیشتر کد نوشته می‌شوند؛ در صورتیکه هر دو روش نتایج یکسانی را تولید می‌کنند.

همچنین به جای استفاده از `==` از `is` و برای `!=` از `isnt` استفاده می‌شود.

```
string = "migrating coconuts"
string == string # true
string is string # true
```

یکی از ویژگی‌های فوق العاده خوب که به CoffeeScript افزوده شده 'or equals' است که با الگو گرفتن از Ruby پیاده سازی شده است.

```
hash or= {}
```

نتیجه‌ی کامپایل آن می‌شود:

```
hash || (hash = {});
```

در اینجا در صورتیکه ارزیابی hash برابر false شود، مقدار آن برابر یک شیء خالی می‌شود. **نکته‌ی مهمی** که وجود دارد در صورتیکه hash مقداری برابر 0، "" و یا null داشته باشد، ارزیابی آن برابر false می‌شود. در صورتی که چنین قصدی ندارید باید از عملگرهای وجودی CoffeeScript استفاده کنید که تنها در حالیکه hash برابر null و یا undefined باشد، فعال می‌شوند.

```
hash ?= {}
```

نتیجه‌ی کامپایل آن می‌شود:

```
if (typeof hash !== "undefined" && hash !== null) {
  hash;
} else {
  hash = {};
};
```