

CoffeeScript #3	عنوان:
وحید محمدطاهری	نویسنده:
۲۱:۴۰ ۱۳۹۴/۰۳/۲۸	تاریخ:
www.dotnettips.info	آدرس:
JavaScript, CoffeeScript	گروه‌ها:

Syntax Object & Array

برای تعریف Object در CoffeeScript می‌توان دقیقاً مانند جاوااسکریپت عمل کرد؛ با یک جفت براکت و ساختار کلید / مقدار. البته همانند تابع، نوشتن براکت اختیاری است. در واقع، شما می‌توانید از تورفتگی و هر کلید/مقدار، در خط جدید به جای کاما استفاده کنید:

```
object1 = {one: 1, two: 2}

# Without braces
object2 = one: 1, two: 2

# Using new lines instead of commas
object3 =
  one: 1
  two: 2

User.create(name: "Vahid Mohammad Taheri")
```

به همین ترتیب، برای تعریف آرایه‌ها می‌توانید از کاما به عنوان جدا کننده و یا هر مقدار آرایه را در یک خط جدید وارد کنید؛ هر چند براکت [] هنوز هم مورد نیاز است.

```
array1 = [1, 2, 3]

array2 = [
  1
  2
  3
]

array3 = [1,2,3,]
```

Flow control

طبق قاعده‌ای که برای نوشتن پرانتز در قبل گفته شد (پرانتز اختیاری است)، در دستورات if و else نیز چنین است:

```
if true == true
  "We're ok"

if true != true then "Vahid"

# برابر است با:
# (1 > 0) ? "Yes" : "No!"
if 1 > 0 then "Yes" else "No!"
```

همانطوری که در مثال بالا مشاهده می‌کنید، در صورتی که از if در یک خط استفاده شود باید پس از شرط، کلمه کلیدی then را بنویسید.

CoffeeScript از اپراتورهای شرطی (?:) پشتیبانی نمی‌کند و به جای آن از if / else استفاده کنید. CoffeeScript نیز همانند Ruby امکان نوشتن بدنه شرط را به صورت پسوندی ایجاد کرده است.

```
alert "It's cold!" if 1 < 5
```

به جای استفاده از علامت ! برای منفی سازی شرط، می‌توانید از کلمه‌ی کلیدی *not* استفاده کنید که سبب خوانایی بیشتر کد نوشته شده می‌شود:

```
if not true then "Vahid"
```

CoffeeScript امکان نوشتن خلاصه‌تر *if not* را نیز ایجاد کرده است؛ برای این کار از کلمه‌ی کلیدی *unless* استفاده کنید. معادل مثال بالا:

```
unless true
  "Vahid"
```

همانند *not* که برای خوانایی بالاتر کد به کار می‌رود، CoffeeScript کلمه کلیدی *is* را مطرح کرده‌است که پس از کامپایل به *===* ترجمه می‌شود.

```
if true is 1
  "OK!"
```

برای نوشتن *!==* نیز می‌توان از *is not* استفاده کرد، که شکل خلاصه‌تر آن *isnt* است.

```
if true isnt true
  alert "OK!"
```

همانطوری که در بالا گفته شد، CoffeeScript عملگر *==* را به *===* و *!=* به *!==* تبدیل می‌کند. دلیلی که CoffeeScript این عمل را انجام می‌دهد این است که جاوااسکریپت عمل مقایسه را بر روی نوع و سپس مقدار آن انجام می‌دهد و سبب پیشگیری از باگ در کد نوشته شده می‌شود.

الحاق رشته ها CoffeeScript امکان الحاق رشته‌ها را با استفاده از روش الحاق رشته‌ها در Ruby فراهم کرده است. برای انجام این عمل از *#{} در داخل "* استفاده کنید که در داخل براکت می‌توانید از دستورات مختلف استفاده کنید. برای مثال:

```
favorite_color = "Blue. No, yel..."
question = "Sam: What... is your favorite color?"
Ben: #{favorite_color}
Sam: Wrong!
```

نتیجه‌ی کامپایل کد بالا می‌شود:

```
var favorite_color, question;
favorite_color = "Blue. No, yel...";
question = "Sam: What... is your favorite color?"
Ben: " + favorite_color + "
Sam: Wrong!";
```