عنوان: CoffeeScript #15
نویسنده: وحید محمّدطاهری
تاریخ: ۱۳۹۴/۰۷/۲۱ ۱۲:۵۰
آدرس: www.dotnettips.info
گروه‌ها: JavaScript, CoffeeScript

قسمت‌های اصلاح نشده در ادامه‌ی مطالب قسمت قبل ، به برخی دیگر از معایب طراحی در جاوااسکریپت که در CoffeeScript نیز اصلاح نشده‌اند می‌پردازیم.

استفاده از parseInt

تابع parseInt() در جاوااسکریپت، در صورتیکه یک مقدار رشته‌ای را به آن ارسال کنید و پایه‌ی مناسب آن را تعیین نکنید، نتایج غیره منتظره‌ای (*unexpected*) را باز می‌گرداند . برای مثال:

```
# Returns 8, not 10!
parseInt('010') is 8
```

البته ممکن است شما این کد را در مرورگر خود تست کنید و مقدار 10 را باز گرداند؛ اما این برای همه‌ی مرورگرها یکسان نیست. برای اطمینان از مقدار بازگشتی صحیح، همیشه پایه‌ی آن را تعیین کنید.

```
# Use base 10 for the correct result
parseInt('010', 10) is 10
```

دقت کنید این چیزی نیست که CoffeeScript بتواند برای شما انجام دهد؛ شما فقط یادتان باشد که همیشه پایه‌ی صحیح را در موقع استفاده‌ی از parseInt() تعریف کنید.

Strict mode

Strict mode یکی از قابلیت‌های ECMAScript 5 است که به شما اجازه می‌دهد تا یک برنامه یا تابع جاوااسکریپت را در محیطی محدود اجرا کنید. این محدودیت موجب نمایش بیشتر خطاها و هشدارها نسبت به حالت نرمال می‌شود و به توسعه دهندگان این امکان را می‌دهد تا از نوشتن کدهای غیر قابل بهینه سازی برای اشتباهات رایج جلوگیری کنند.
به عبارت دیگر Strict mode باعث کاهش اشکالات، افزایش امنیت، بهبود عملکرد و حذف برخی از سختی‌های استفاده از ویژگی‌های زبان می‌شود.
در حال حاضر Strict mode، در مرورگرهای زیر پشتیبانی می‌شود:

Chrome >= 13.0
Safari >= 5.0
Opera >= 12.0
Firefox >= 4.0
IE >= 10.0

با این حال، Strict mode به طور کامل با مرورگرهای قدیمی سازگار است.

تغییرات Strict mode

بیشتر تغییرات Strict mode مربوط به syntax جاوااسکریپت بوده است:

خطا در پروپرتی‌ها و نام آرگومان‌های تابع تکراری
خطا در عدم استفاده‌ی صحیح از *delete*
خطا در زمان دسترسی به arguments.caller و arguments.callee (به دلایل عملکرد)

استفاده از عمگر *with* سبب بروز خطای نحوی می‌شود
متغیرهای خاص مانند *undefined* که قابل نوشتن نیستند
معرفی کلمات کلیدی رزرو شده مانند *implements* , *interface* , *let* , *package* , *private* , *protected* , *public* , *static* و *yield* .

با این حال، برخی از رفتارهای زمان اجرای Strict mode نیز تغییر کرده است:
متغییرهای سراسری به صورت صریح و روشن هستند (کلمه کلیدی var نیاز است). مقدار سراسری *this* نیز به صورت undefined است.
eval نمی‌تواند متغیر جدیدی را در حوزه‌ی محلی خود تعریف کند.
بدنه‌ی هر تابع باید قبل از استفاده تعریف شده باشد (قبلا گفتم که در جاوااسکریپت شما می‌توانید قبل از تعریف تابع آن را فراخوانی کنید).
آرگومان‌ها تغییر ناپذیر هستند.

CoffeeScript در حال حاضر بسیاری از الزامات Strict mode را پیاده سازی کرده‌است مانند: همیشه از کلمه کلیدی var برای تعریف متغیر استفاده می‌کند؛ اما فعال کردن Strict mode در برنامه‌های CoffeeScript نیز بسیار مفید خواهد بود. در واقع CoffeeScript بر روی انطباق برنامه‌ها با Strict mode در زمان کامپایل را، در برنامه‌های آینده خود دارد.

استفاده از Strict mode
برای فعال کردن بررسی محدودیت، کد و توابع خود را با این رشته شروع کنید:

```
->
  "use strict"

  # ... your code ...
```

فقط با استفاده از رشته "use strict". به مثال زیر توجه کنید:

```
do ->
  "use strict"
  console.log(arguments.callee)
```

اجرای قطعه کد بالا درحالت strict mode، سبب بروز خطای syntax می‌شود؛ در حالیکه در حالت معمول این کد به خوبی اجرا می‌شود.
Strict mode دسترسی به arguments.callee و arguments.caller، که تاثیر بدی را بر روی عملکرد کد شما دارند، حذف می‌کند و استفاده‌ی از آنها سبب بروز خطا می‌شود.

در مثال زیر در حالت strict mode سبب بروز خطای *TypeError* می‌شود، اما در حالت نرمال به خوبی اجرا شده و یک متغیر سراسری را ایجاد می‌کند.

```
do ->
  "use strict"
  class @Spine
```

دلیل این رفتار این است که در Strict mode متغیر *this* به صورت *undefined* است؛ در حالیکه در حالت نرمال، *this* به شیء *window* اشاره می‌کند. راه حل این مشکل تعریف متغیرهای سراسری به صورت صریح به شیء window است.

```
do ->
  "use strict"
  class window.Spine
```

در حالیکه توصیه می‌شود که همیشه Strict mode فعال باشد، اما Strict mode هیچ یک از ویژگی‌های جدید جاوااسکریپت را که

هنوز آماده نیست، فعال نمی‌کند و در واقع به علت بررسی بیشتر کدهای شما در زمان اجرا، باعث کاهش سرعت می‌شود.
شما می‌توانید در زمان توسعه برنامه جاوااسکریپت خود Strict mode را فعال کنید و در زمان انتشار، بدون Strict mode برنامه‌ی خود را منتشر کنید.

JavaScript Lint

JavaScript Lint یک ابزار بررسی کیفیت کدهای جاوااسکریپت است و اجرای برنامه‌ی شما از طریق این راه عالی باعث بهبود کیفیت و بهترین شیوه‌ی کد نویسی می‌شود. این پروژه براساس ابزار JSLint است. شما می‌توانید چک لیست سایت JSLint را که شامل موضوعاتی است که باید آن‌ها در نظر داشته باشید، مانند متغیرهای سراسری، فراموش کردن نوشتن سمی کالن، کیفیت ضعیف عمل مقایسه را نام برد.

خبر خوب این است که CoffeeScript تمام موارد گفته شده‌ی در چک لیست را انجام می‌دهد. بنابراین کد تولیدی CoffeeScript با JavaScript Lint کاملا سازگار است. در واقع ابزار *coffee* از option، lint پشتیبانی می‌کند.

```
coffee --lint index.coffee
  index.coffee: 0 error(s), 0 warning(s)
```