

در ابتدا به توضیحاتی درباره کنترل نوع به صورت ایستا و کنترل نوع در زمان اجرا، توجه کنید:  
**کنترل نوع ایستا** ( Static Type Checking ) کامپایلر را قادر به بررسی درستی برنامه می‌کند، بدون آنکه آن را اجرا کند. مثلاً کد زیر با خطا مواجه می‌شود:

```
int x = "5";
```

**کنترل نوع در زمان اجرا** (RunTime Type Checking)، هنگامی که برنامه اجرا می‌شود این کنترل توسط [CLR](#) صورت می‌گیرد و موقع تایپ کد، خطایی گرفته نمی‌شود مثلاً:

```
object y = "5";  
int z = (int) y; //downcast شکست خطای زمان اجرا و شکست
```

[downcast](#) به عمل تبدیل نوع کلاس پایه به یکی از کلاس‌های مشتق شده، گفته می‌شود.  
 پس از ذکر مقدمه بالا به این سؤال می‌پردازیم که تفاوت انواع var و dynamic چیست؟  
 کلمه کلیدی var و کلمه کلیدی dynamic علی‌الظاهر کاربرد یکسانی دارند اما تفاوت اساسی آن‌ها عبارت است از این‌که نوع واقعی متغیرهایی از نوع var، توسط کامپایلر تعیین می‌شود. یعنی متغیرهایی از نوع var کنترل نوعشان به صورت ایستاست اما نوع واقعی متغیرهایی از نوع dynamic، در زمان اجرا مشخص می‌شود. یعنی متغیرهایی از نوع dynamic کنترل نوعشان به صورت کنترل نوع در زمان اجراست. به کد زیر توجه کنید:

```
dynamic x = "hello"; // این متغیر در زمان کامپایل از نوع دینامیک است و در زمان اجرا از نوع رشته‌ای است  
var y = "hello"; // این متغیر هم در زمان کامپایل و هم در زمان اجرا از نوع رشته‌ای است  
int i = x; // خطای زمان اجرا  
int j = y; // خطای زمان کامپایل
```

همچنین متغیری که نوعش var است می‌تواند مقداری از نوع dynamic را شامل شود.

```
dynamic x = "hello";  
var y = x; // این متغیر در زمان کامپایل دینامیک است  
int z = y; // خطای زمان اجرا
```

## نظرات خوانندگان

نویسنده: محسن  
تاریخ: ۱۳۹۱/۰۷/۱۲ ۸:۳۱

تفاوت مهم دیگه‌ای که بین این دو وجود دارد این است که می‌توان نوع تعریف شده توسط dynamic را در حین کدنویسی تغییر داد اما اینکار برای var امکان پذیر نیست (با انتساب اول نوع برای متغیر در نظر گرفته می‌شود. برای مثال:

```
int k=100;
var data1=k;
dynamic data2=k;
data1="Hello World!"/>

```

خطای حین کامپایل به دلیل تغییر نوع  
بدون خطا در زمان کامپایل و یا اجرا

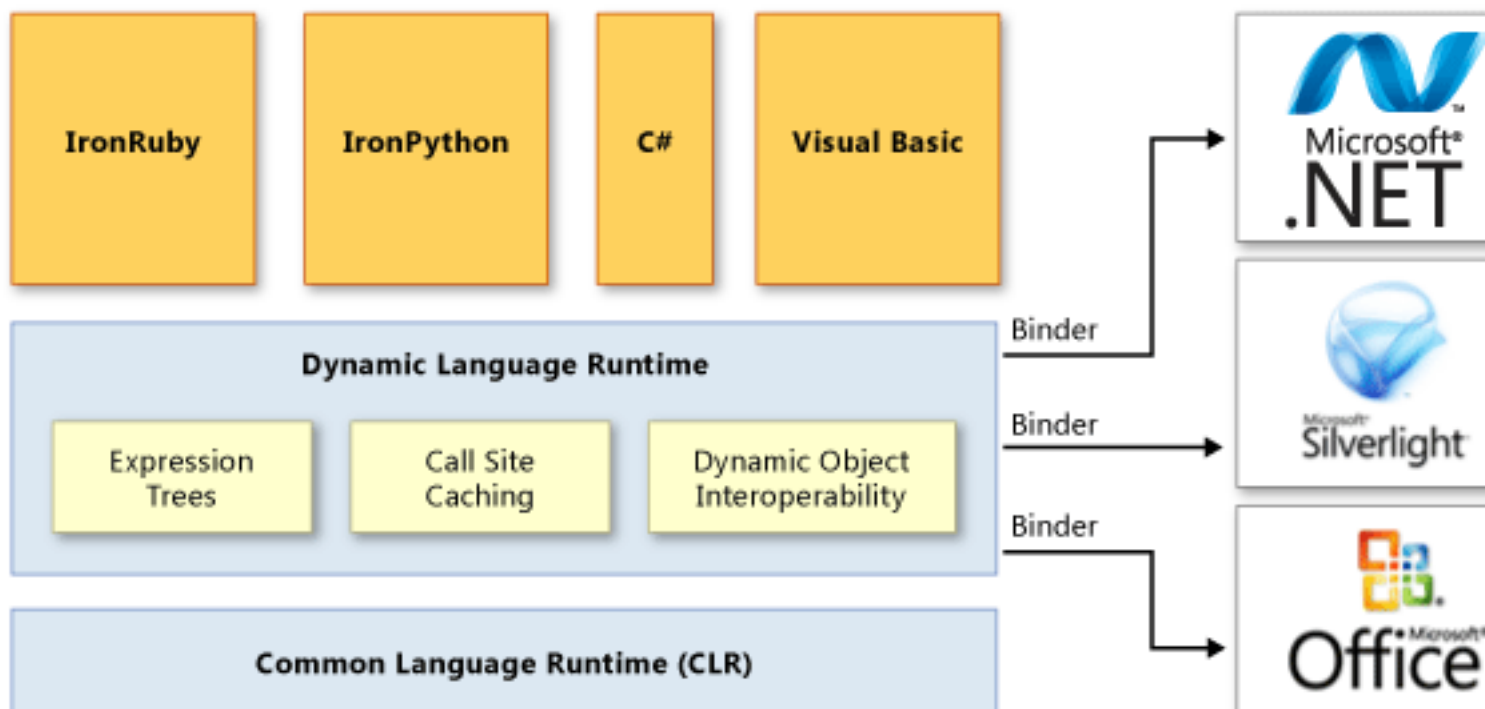
نویسنده: علیرضا صالحی  
تاریخ: ۱۳۹۱/۰۷/۱۲ ۱۰:۳۲

کارکرد کلمه کلیدی var تحت عنوان Type Inference (استنتاج نوع) شناخته می‌شود. این اصطلاح به این معنی است که در زمان کامپایل نوع متغیر قابل تشخیص است و صرفاً یک راهکار برای سادتر شدن کار برنامه نویسی است، در عمل و از دید کامپایلر این همان تعریف متغیر معمولی (statically typed) است.

[Type inference](#)

از آنجایی که DLR بر روی CLR پیاده شده است dynamic در واقع خودش استاتیک است! اما قابلیت‌های دینامیک بودن را ارائه می‌کند.

statically typed dynamic



نویسنده: وحید نصیری  
تاریخ: ۱۳۹۱/۰۷/۱۲ ۱۲:۲۷

[DLR](#) از CLR که مساوی (GC/JIT/sandboxed security model/etc) هست استفاده می‌کند اما فقط از سرویس‌های آن [و نه اینکه](#) ذاتاً و نهایتاً استاتیک کار کند. DLR یک opcode جدید را به این مجموعه اضافه می‌کند به نام [InvokeDynamic](#) که پیشتر در CLR [موجود نبوده](#) .

نویسنده: علیرضا صالحی  
تاریخ: ۱۳۹۱/۰۷/۱۲ ۱۶:۵۲

بله اینها صحیح، ولی statically typed dynamic عبارتی است که Anders Hejlsberg هنگام صحبت در مورد آینده #C در PDC09 به کاربرد.

[#The Future of C](#)

The dynamic keyword acts as a static type declaration in the C# type system. This way C# got the dynamic features and at the same time remained a statically typed language

<http://msdn.microsoft.com/en-us/magazine/gg598922.aspx>

در واقع کلمه کلیدی dynamic به کامپایلر می‌فهماند که compile-time checking را غیر فعال کن! تا در زمان اجرا به نوع متغییر رسیدگی شود.

شاید اگر بگوییم dynamic نوعی static است که مزایای انواع dynamic را در بر می‌گیرد بهتر باشد.

خواندن این مقاله هم خالی از لطف نیست:

[Static Typing Where Possible, Dynamic Typing When Needed: The End of the Cold War Between Programming Languages](#)