

## آشنایی با RLE

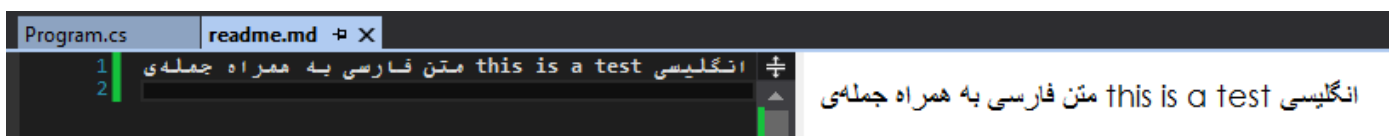
[الگوریتم پردازش دوطرفه‌ی یونیکد](#)، جهت و سمت نمایش متن را بر اساس خواص جهتی هر حرف مشخص می‌کند. در این حالت اگر متن مورد نمایش، انگلیسی و یا فارسی خالص باشند به خوبی عمل می‌کند؛ اما اگر ترکیب این دو را در یک رشته داشته باشیم، نیاز است نحوه‌ی جهت گیری و نمایش حروف را به Unicode bidirectional algorithm معرفی کنیم. این نوع مشکلات را فارسی زبان‌ها در حین نمایش ترکیبی از متن فارسی و انگلیسی در Tooltips، برنامه‌های نمایش زیرنویس‌های فیلم‌ها، برنامه‌های گزارشگیری و امثال آن به وفور مشاهده می‌کنند. راه حل استاندارد یونیکد آن، استفاده از حروف نامرئی یونیکد است که جهت نمایشی متن جاری را بازنویسی می‌کنند:

```
U+202A: LEFT-TO-RIGHT EMBEDDING (LRE)
U+202B: RIGHT-TO-LEFT EMBEDDING (RLE)
U+202D: LEFT-TO-RIGHT OVERRIDE (LRO)
U+202E: RIGHT-TO-LEFT OVERRIDE (RLO)
U+202C: POP DIRECTIONAL FORMATTING (PDF)
```

برای مثال حرف یونیکد نامرئی U202B به این معنا است: «از این لحظه به بعد تا اطلاع ثانوی، متن نمایش داده شده راست به چپ است؛ صرفنظر از خواص جهتی حروف مورد استفاده». این تا اطلاع ثانوی یا POP نیز توسط حرف U202C مشخص شده و به پایان می‌رسد. به عبارتی یونیکد شبیه به یک پشته یا Stack عمل می‌کند.

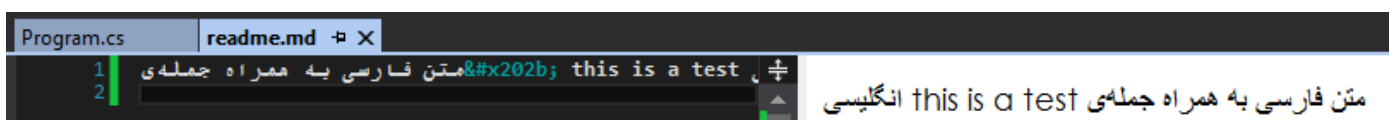
## مثال اول

عبارت «متن فارسی به همراه جمله‌ی this is a test انگلیسی» را در نظر بگیرید. اکنون فرض کنید می‌خواهیم از آن جهت ارائه یک فایل readme مخصوص GitHub با فرمت mark down یا md استفاده کنیم:



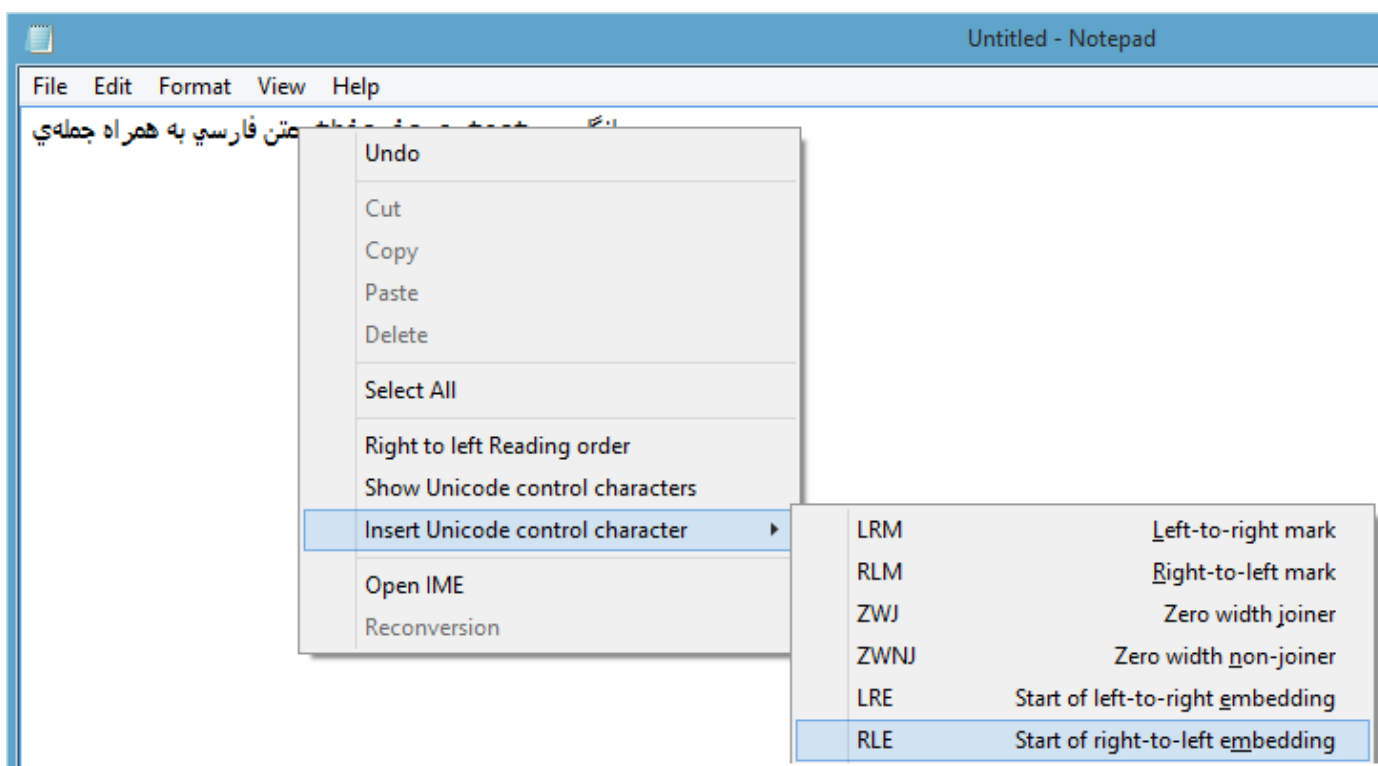
همانطور که ملاحظه می‌کنید، جمله معکوس شده است. برای رفع این مشکل می‌توان از کاراکتر نامرئی یونیکد 202b استفاده کرد. البته در mark down امکان تعریف ساده‌تر این کاراکتر به صورت ذیل نیز پیش بینی شده است:

```
&#x202b;
```

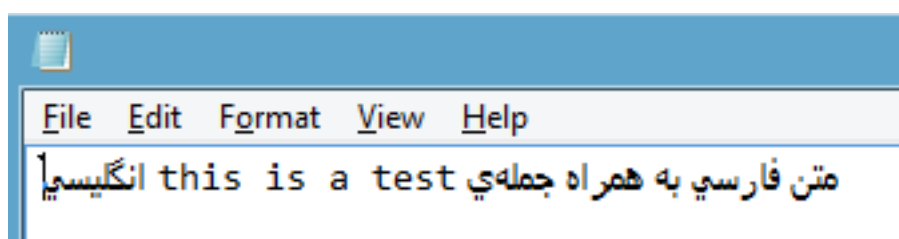


## مثال دوم

اغلب نمایشگرهای چپ به راست متون نیز در حالت پیش فرض، عبارت مثال اول را معکوس نمایش می‌دهند:



اگر از notepad استفاده کنید، به صورت توکار امکان افزودن RLE را به ابتدای جمله دارد:



## مثال سوم

در زبان‌های دات نت نیز جهت نمایش صحیح متون ترکیبی، می‌توان حرف RLE را به صورت ذیل به ابتدای یک جمله اضافه کرد:

```
public const char RightToLeftEmbedding = (char)0x202B;
```

این مورد خصوصاً در ابزارهای گزارشگیری یا کار با API ویندوز می‌تواند مفید باشد.

تشخیص راست به چپ بودن متن

در محیط وب جهت نمایش صحیح یک متن نیز می‌توان به مرورگرها کمک کرد. تعریف `dir=rtl` تفاوتی با قرار دادن RLE در ابتدای یک متن ندارد. در این حالت نیاز است بدانیم حروف RTL در چه بازه‌ای از شماره حروف یونیکد قرار می‌گیرند:

Right-to-left Unicode blocks for modern scripts are:

Consecutive range of the main letters:

U+0590 to U+05FF - Hebrew  
U+0600 to U+06FF - Arabic  
U+0700 to U+074F - Syriac  
U+0750 to U+077F - Arabic Supplement  
U+0780 to U+07BF - Thaana  
U+07C0 to U+07FF - N'Ko  
U+0800 to U+083F - Samaritan

Arabic Extended:

U+08A0 to U+08FF - Arabic Extended-A

Consecutive presentation forms:

U+FB1D to U+FB4F - Hebrew presentation forms  
U+FB50 to U+FDFF - Arabic presentation forms A

More Arabic presentation forms:

U+FE70 to U+FEFF - Arabic presentation forms B

که یک نمونه‌ی ساده شده‌ی این بازه‌ها، به صورت ذیل است:

```
private static readonly Regex _matchArabicHebrew =  
new Regex(@"[\u0600-\u06FF,\u0590-\u05FF]", RegexOptions.IgnoreCase | RegexOptions.Compiled);  
  
public static bool ContainsRtlFarsi(this string txt)  
{  
    return !string.IsNullOrEmpty(txt) && _matchArabicHebrew.IsMatch(txt);  
}
```

و حالت پیشرفته‌تر آن‌را که سایت توئیتر برای ارائه‌ی یک جعبه متنی به صورت خودکار راست به چپ شونده، مورد استفاده قرار می‌دهد، در اینجا می‌توانید مطالعه کنید:

[RTLText.module.js](#)

نمایش صحیح عبارات ممیز دار در یک گزارش راست به چپ

تاریخ: 18/11/1390  
شماره پروژ: 56/4/3/2/1  
اسلش: A/13/12  
بك اسلش: 14\13\12  
مساوي و جمع: 5=3+2  
سمي كولون: ;1+1=2  
دلار: \$12  
كاما: 12,34,67  
نقطه: 12.34  
پرانتز: متن (ساده)

استاندارد یونیکد یک سری کاراکتر را «کاراکتر ضعیف» معرفی کرده‌است. برای مثال کاراکتر اسلش بکار رفته در یک تاریخ هم از این دست است. بنابراین اگر در یک گزارش تولیدی، شماره کد ممیز دار و یا یک تاریخ را معکوس مشاهده می‌کنید به این علت است که یک «نویسه ضعیف» مثل اسلش نمی‌تواند جهت را تغییر دهد؛ مگر اینکه از یک «نویسه قوی» برای دستکاری آن استفاده شود (مانند RLE و POP که در ابتدای بحث معرفی شدند).

یک مطلب تکمیلی در این مورد: «[iTextSharp و نمایش صحیح تاریخ در متنی راست به چپ](#)»  
این اصول در تمام محیط‌هایی که از یونیکد پشتیبانی می‌کنند صادق است و تفاوتی نمی‌کند که ویندوز باشد یا Adobe reader و یا یک ابزار گزارش‌گیری که اصلاً برای محیط‌های راست به چپ طراحی نشده‌است.

### کار با اعراب در متون راست به چپ

در یونیکد یک حرف می‌تواند از یک یا چند code point تشکیل شود. در حالت FormC، هر حرف، با اعراب آن یک code point را تشکیل می‌دهند. در حالت FormD، حرف با اعراب آن دو code point را تشکیل خواهند داد. به همین جهت نیاز است رشته را تبدیل به حالت D کرد تا بتوان اعراب آن‌را مجزای از حروف پایه، حذف نمود.  
البته اعراب در اینجا به اعراب عربی ختم نمی‌شود. یک سری حروف اروپایی مانند "ö", "ä", و "ü" را نیز شامل می‌شود.  
یک مطلب تکمیلی در این مورد: «[حذف اعراب از حروف و کلمات](#)»

## نظرات خوانندگان

نویسنده:

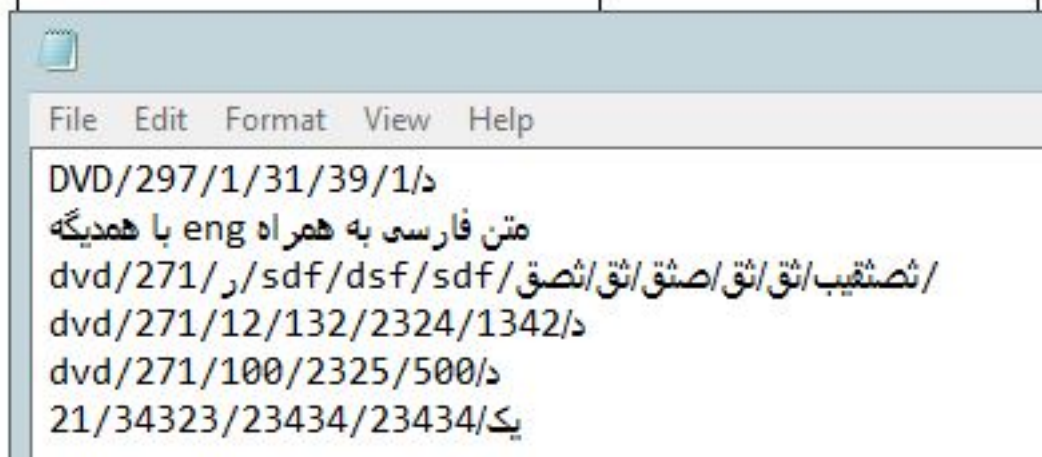
آقا ابراهیم

## تاریخ:

17:38 1393/10/16

سلام. بسیار استفاده بردیم. اما یک سوال؛ من دیتایی مثل تصویر زیر دارم. اما وقتی اعداد وارد میکنم و اسلش میزنم، رشته به هم میریزه:

DVD/297/1/د/1/1/61	درسی
CD/297/1/د/1/1/62	درسی
DVD/297/1/د/26/11/4	درسی
CD/297/2/د/1/1/63	درسی
CD/297/1/د/32/36/1	درسی



من می‌خواهم مثل فایل ورد باشه و همه چیز سر جاش. اما وقتی همون فایل ورد کپی می‌گیرم داخل Notpad به هم می‌زنه. از روش شما استفاده کردم. تونستم جمله‌ی فارسی+انگلیسی بنویسم. البته من می‌خواهم اول بنویسم 12/452/121/CharFarsi/214/dvd. اما همیشه اون بخش CharFarsi میره به آخر. ممنون میشم بهم یاد بدید که چطوری از کاربر بگیرم که به هم نریزه و حتی وقتی سرچ میکنم رشته رو بدون مشکل بیداش کنم.

نویسنده:

## وحید نصیری

تاریخ:

17:49 1393/10/16

در مطلب « [iTextSharp و نمایش صحیح تاریخ در متنی راست به چپ](#) » متد FixWeakCharacters، برای رفع این مشکل در حین تهیه گزارش‌های PDF ای، تهیه شد:

```
const char RightToLeftEmbedding = (char)0x202B;
const char PopDirectionalFormatting = (char)0x202C;

static string FixWeakCharacters(string data)
```

```
{
    if (string.IsNullOrEmpty(data)) return string.Empty;
    var weakCharacters = new[] { @"\", "/", "+", "-", "=", ";", "$" };
    foreach (var weakCharacter in weakCharacters)
    {
        data = data.Replace(weakCharacter, RightToLeftEmbedding + weakCharacter +
PopDirectionalFormatting);
    }
    return data;
}
```

اگر از این متد استفاده نشود، دقیقاً خروجی نمایشی PDF اسلش دار، با خروجی نوت پدی که ارائه دادید یکی خواهد بود. بنابراین همین متد را باید در رخداد on key press و امثال آن، جهت اصلاح جهت ورود کاراکترها فراخوانی کنید. البته این را هم در نظر داشته باشید که برای مثال RLE/POP ایی که در این متد به صورت خودکار درج می‌شود، برای نمایش نهایی طراحی شده‌است (استفاده برای یکبار) و اگر قرار است در on key press فراخوانی شود باید بررسی کنید که آیا قبلاً RLE/POP را درج کرده‌اید یا خیر. همچنین بدیهی است در حین جستجو باید RLE و POP را از رشته‌ی دریافتی حذف کنید (یک Replace ساده با string.Empty)