

عنوان: تعریف رنگ در iTextSharp

نویسنده: وحید نصیری

تاریخ: ۱۳۹۰/۰۶/۰۲ ۱۹:۳۱:۰۰

آدرس: www.dotnettips.info

برچسب‌ها: iTextSharp

در کتابخانه‌ی iTextSharp به جهت سازگاری با کتابخانه‌ی اصلی، رنگ‌ها را بر اساس کلاسی به نام BaseColor تعریف کرده‌اند؛ که ای‌کاش به جای این‌کار، همه را با کلاس Color فضای نام استاندارد System.Drawing جایگزین می‌کردند. همین مشکل با فونت هم هست. یک کلاس فونت در فضای نام iTextSharp.text وجود دارد به علاوه کلاس فونت تعریف شده در فضای نام استاندارد System.Drawing. دات‌نت؛ که خیلی سریع می‌تواند به خطای کامپایل زیر ختم شود:

'Font' is an ambiguous reference between 'iTextSharp.text.Font' and 'System.Drawing.Font'

و در نهایت مجبور خواهیم شد که به صورت صریح علام کنیم، iTextSharp.text.Font منظور ما است و نه آن یکی. در کل اگر با کلاس Color فضای نام استاندارد System.Drawing بیشتر راحت هستید به صورت زیر هم می‌توان رنگ‌های متداول را مورد استفاده قرار داد:

تعریف رنگ‌ها بر اساس نام آن‌ها:

```
var color = new BaseColor(Color.LightGray);
```

تعریف رنگ‌ها بر اساس مقادیر Hex متداول در المان‌های HTML :

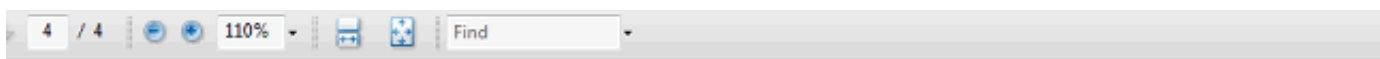
```
var color = new BaseColor(ColorTranslator.FromHtml("#1C5E55"));
```


این نکته‌ای است که شاید خیلی‌ها از وجود آن بی‌اطلاع باشند. به صورت پیش فرض در کلاس استاندارد ColorTranslator، امکان دریافت رنگ‌های بکاررفته در المان‌های HTML به کمک متد ColorTranslator.FromHtml مهیا است. البته اگر زمانی خواستید خودتان این متد را پیاده سازی کنید، نکته‌ی آن به صورت زیر است:

```
string htmlColor = "#1C5E55";
int x = Convert.ToInt32(htmlColor.Replace("#", "0x"), 16);
byte red = (byte)((x & 0xff0000) >> 16);
byte green = (byte)((x & 0xff00) >> 8);
byte blue = (byte)(x & 0xff);
```

سپس کلاس‌های Color و همچنین BaseColor امکان پذیرش این اجزای حاصل را دارند (به کمک متد Color.FromRgb یا سازنده‌ی BaseColor). علت ذکر ColorTranslator.FromHtml به این بر می‌گردد که ترکیبات رنگ جالبی را می‌توان از جداول HTML ای‌بی موجود در

سایت‌های مختلف ایده گرفت و یا حتی از قالب‌های پیش فرض GridView در ASP.NET مثلاً.



 <p>گزارش جدید ما</p>				
ردیف	شماره	نام	نام خانوادگی	موجودی
			نقل از صفحه قبل	۶۲,۴۵۰
۱۳۳	۱۳۲	نام ۱۳۲	نام خانوادگی ۱۳۲	۲۰۹
۱۳۴	۱۳۳	نام ۱۳۳	نام خانوادگی ۱۳۳	۶۰۳
۱۳۵	۱۳۴	نام ۱۳۴	نام خانوادگی ۱۳۴	۷۳۴
۱۳۶	۱۳۵	نام ۱۳۵	نام خانوادگی ۱۳۵	۹۶۵
۱۳۷	۱۳۶	نام ۱۳۶	نام خانوادگی ۱۳۶	۶
			جمع صفحه	۲,۵۱۷
			جمع کل	۶۴,۹۶۷

اکنون برای ساخت جدولی مانند شکل فوق، به ازای هر سلولی که مشاهده می‌کنید باید یکبار `BorderColor` و `BackgroundColor` تنظیم شوند. رنگ متن هم از رنگ فونت دریافت می‌شود:

```
var pdfCell = new PdfPCell(new Phrase(Text, Font))
{
    RunDirection = ...,
    BorderColor = ...,
    BackgroundColor = ...
};
```

نظرات خوانندگان

نویسنده: Mohsen Najafzadeh
تاریخ: ۱۰:۲۰:۴۴ ۱۳۹۰/۰۶/۰۶

سلام استاد
اگه راه داره این مثال رو رو سایت قرار بدین ممنون می شم

نویسنده: وحید نصیری
تاریخ: ۱۰:۵۲:۴۹ ۱۳۹۰/۰۶/۰۶

یک مثال خوب در این زمینه: [تهیه خروجی PDF از GridView](#)

نویسنده: Mohsen Najafzadeh
تاریخ: ۱۱:۰۱:۳۴ ۱۳۹۰/۰۶/۰۶

ممنون
یا علی

نویسنده: Nima
تاریخ: ۰۰:۵۴:۲۶ ۱۳۹۰/۰۶/۱۳

ممنون از مثال مفیدتون
میخواستم بپرسم چطور میتونم عکس رو در کنار متن همینطوری که شما در عکس بالا دارین داشته باشم؟ من به خاصیت Image یک سلول وقتی مقدار میدم کل سلول رو میگیره و وقتی که تو تا سلول تعریف میکنم Border ها در دسر ساز میشن و من میخوام دقیقا مثل عکسی که در بالا هست بدون Border باشه

نویسنده: وحید نصیری
تاریخ: ۰۱:۱۰:۳۹ ۱۳۹۰/۰۶/۱۳

هدر مثال بالا یک جدول است (یک ستون دارد و سه ردیف). عکس هم در یک سلول آن (PdfPCell) قرار گرفته (اولین سلول که می شود ردیف یک). برای عدم نمایش حاشیه یا border ، مقدار آن را مساوی صفر قرار دهید (cell.Border = 0). برای اینکه کل سلول را پوشش ندهد خاصیت fit image را مانند مثالی که قبلا در این سایت زده شده، false کنید. برای اینکه در وسط قرار گیرد، cell.HorizontalAlignment = 1 را تنظیم کنید.