

کتابخانه iTextSharp نمایش گرادیانی از رنگ‌ها را هم پشتیبانی می‌کند و بدیهی است این نمایش برداری است. روش استفاده از آن هم بسیار ساده است؛ مثلاً:

```
PdfShading shading = PdfShading.SimpleAxial(pdfWriter, x0, y0, x1, y1, BaseColor.YELLOW,
BaseColor.RED);
PdfShadingPattern pattern = new PdfShadingPattern(shading);
ShadingColor color = new ShadingColor(pattern);
```

متد PdfShading.SimpleAxial بر اساس شیء PdfWriter که توسط آن به Canvas صفحه دسترسی پیدا می‌کند، در مختصاتی مشخص، یک طیف رنگی را ایجاد می‌کند. بر این اساس می‌توان به یک ShadingColor هم رسید که از آن مثلاً به عنوان BackgroundColor یک PdfPCell قابل استفاده است. تا اینجا ساده به نظر می‌رسد اما واقعیت این است که مختصات ذکر شده، مهم است و آن‌را در مورد مثلاً سلول‌های یک جدول تنها در زمان تهیه نهایی یک جدول می‌توان به دست آورد. البته اگر شیء ساده‌ای را روی صفحه رسم کنیم، این مورد بر اساس مختصات ابتدایی شیء واضح به نظر می‌رسد. برای حل این مشکل در مورد جداول و سلول‌های آن خاصیتی به نام CellEvent وجود دارد که از نوع IPdfPCellEvent است. به عبارتی با ارسال یک وهله از کلاسی که اینترفیس IPdfPCellEvent را پیاده سازی می‌کند، می‌توان در زمان نمایش نهایی یک سلول به مختصات دقیق آن دسترسی پیدا کرد. یک مثال کامل را در مورد پیاده سازی IPdfPCellEvent و استفاده از آن جهت نمایش گرادیان در Header و footer یک جدول، در ادامه مشاهده خواهید نمود:

```
using System.Diagnostics;
using System.IO;
using iTextSharp.text;
using iTextSharp.text.pdf;

namespace iTextSharpGradientTest
{
    public class GradientCellEvent : IPdfPCellEvent
    {
        public void CellLayout(PdfPCell cell, Rectangle position, PdfContentByte[] canvases)
        {
            var cb = canvases[PdfPTable.BACKGROUNDCANVAS];
            cb.SaveState();

            var shading = PdfShading.SimpleAxial(
                cb.PdfWriter,
                position.Left, position.Top, position.Left, position.Bottom,
                BaseColor.YELLOW, BaseColor.ORANGE);
            var shadingPattern = new PdfShadingPattern(shading);

            cb.SetShadingFill(shadingPattern);
            cb.Rectangle(position.Left, position.Bottom, position.Width, position.Height);
            cb.Fill();

            cb.RestoreState();
        }
    }

    class Program
    {
        static void Main(string[] args)
        {

```

```

using (var pdfDoc = new Document(PageSize.A4))
{
    var pdfWriter = PdfWriter.GetInstance(pdfDoc, new FileStream("Test.pdf",
        FileMode.Create));
    pdfDoc.Open();

    var table1 = new PdfPTable(1);
    table1.HeaderRows = 2;
    table1.FooterRows = 1;

    //header row
    var headerCell = new PdfPCell(new Phrase("header"))
    {
        HorizontalAlignment = Element.ALIGN_CENTER,
        Border = 0
    };
    headerCell.CellEvent = new GradientCellEvent();
    table1.AddCell(headerCell);

    //footer row
    var footerCell = new PdfPCell(new Phrase("footer"))
    {
        HorizontalAlignment = Element.ALIGN_CENTER,
        Border = 0
    };
    footerCell.CellEvent = new GradientCellEvent();
    table1.AddCell(footerCell);

    //adding some rows
    for (int i = 0; i < 70; i++)
    {
        var rowCell = new PdfPCell(new Phrase("Row " + i)) { BorderColor =
BaseColor.LIGHT_GRAY };
        table1.AddCell(rowCell);
    }

    pdfDoc.Add(table1);
}

//open the final file with adobe reader for instance.
Process.Start("Test.pdf");
}
}

```

با خروجی:

header
Row 0
Row 1
Row 2
Row 3
Row 4

نکته مهم این مثال نحوه مقدار دهی CellEvent است. به این ترتیب در زمان نمایش نهایی یک سلول می‌توان در متد CellLayout، به خواص فقط خواندنی آن سلول دسترسی یافت. مثلاً position، مختصات نهایی مستطیل مرتبط با سلول جاری را بر می‌گرداند؛ یا از طریق canvases می‌توان برای آخرین بار فرصت یافت تا در یک سلول نقاشی کرد.

## نظرات خوانندگان

نویسنده: alireza

تاریخ: ۱۳۹۰/۰۹/۱۸ ۰۹:۲۸

با سلام و خسته نباشید ، از این کلاس من در سایت خودم برای تبدیل مقالات به فرمت pdf استفاده کردم متون فارسی را تبدیل به حروف ناخوانا می کند در این آدرس این مشکل را می توانید مشاهده نمایید :  
<http://article.kiansoftware.com/article.aspx?id=117&idauthore=1>  
راه حلش رو اگر توضیح بفرمایید ممنون می شم . با تشکر

نویسنده: وحید نصیری

تاریخ: ۱۳۹۰/۰۹/۱۹ ۰۹:۴۹:۴۲

لطفا مراجعه کنید به برچسب iTextSharp که در کنار صفحه هست. چند مقاله در این مورد منتشر کردم:

[فارسی نویسی و iTextSharp](#)

[روش صحیح تعریف قلم در iTextSharp](#)

[تبدیل HTML به PDF با استفاده از کتابخانه iTextSharp](#)

[iTextSharp و استفاده از قلمهای محدود فارسی](#)