

روش متداول کار با کتابخانه‌ی iTextSharp ، ایجاد شیء Document ، سپس ایجاد PdfWriter برای نوشتن در آن، گشودن سند و ... افزودن اشیایی مانند PdfPTable ، PdfPCell و Paragraph و غیره به آن است و در نهایت بستن سند. راه میانبری هم برای کار با این کتابخانه وجود دارد و آن هم استفاده از امکانات فضای نام iTextSharp.text.html.simpleparser آن می‌باشد. به این ترتیب می‌توان به صورت خودکار، یک محتوای HTML را تبدیل به فایل PDF کرد.

مثال : نمایش یک متن HTML ساده انگلیسی

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

namespace HeadersAndFooters
{
    class Program
    {
        static void Main(string[] args)
        {
            using (var pdfDoc = new Document(PageSize.A4))
            {
                PdfWriter.GetInstance(pdfDoc, new FileStream("Test.pdf", FileMode.Create));
                pdfDoc.Open();

                var html = @"<span style='color:blue'><b>Testing</b></span>
                            <i>iTextSharp's</i> <u>HTML to PDF capabilities</u>";
                var parsedHtmlElements = HTMLWorker.ParseToList(new StringReader(html), null);

                foreach (var htmlElement in parsedHtmlElements)
                {
                    pdfDoc.Add(htmlElement);
                }
            }

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

```

    }
}
}

```

نکته‌ی جدید کد فوق، استفاده از متد `HTMLWorker.ParseToList` است. به این ترتیب parser کتابخانه‌ی iTextSharp وارد عمل شده و html تعریف شده را به معادل المان‌های بومی خودش تبدیل می‌کند؛ مثلاً تبدیل به `chunk` یا `pdfpTable` و امثال آن. در نهایت در طی یک حلقه، این عناصر به صفحه اضافه می‌شوند.

البته باید دقت داشت که HTMLWorker امکان تبدیل عناصر پیچیده، تودرتو و چندلایه HTML را ندارد؛ اما بهتر از هیچی است!

همه‌ی این‌ها خوب! اما به درد ما فارسی زبان‌ها نمی‌خورد. همین متغیر `html` فوق را با یک متن فارسی جایگزین کنید، چیزی نمایش داده نخواهد شد. البته این هم نکته دارد که در ادامه ذکر خواهد شد.

جهت نمایش متون فارسی نیاز است تا نکات ذکر شده در مطلب «[فارسی نویسی و iTextSharp](#)» رعایت شوند که شامل:

- تعیین صریح قلم

- تعیین encoding

- استفاده از عناصر دربرگیرنده‌ای است که خاصیت `RunDirection` را پشتیبانی می‌کنند؛ مانند `PdfPCell` و غیره

به این ترتیب خواهیم داشت:

```

using System.Diagnostics;
using System.IO;
using iTextSharp.text;
using iTextSharp.text.html.simpleparser;
using iTextSharp.text.pdf;
using iTextSharp.text.html;

namespace HeadersAndFooters
{
    class Program
    {
        static void Main(string[] args)
        {
            using (var pdfDoc = new Document(PageSize.A4))
            {
                PdfWriter.GetInstance(pdfDoc, new FileStream("Test.pdf", FileMode.Create));
                pdfDoc.Open();

                // روش صحیح تعریف فونت
                FontFactory.Register("c:\\windows\\fonts\\tahoma.ttf");

                StyleSheet styles = new StyleSheet();
                styles.LoadTagStyle(HtmlTags.BODY, HtmlTags.FONTFAMILY, "tahoma");
            }
        }
    }
}

```

```

styles.LoadTagStyle(HtmlTags.BODY, HtmlTags.ENCODING, "Identity-H");

var html = @"<span style='color:blue'><b>آزمایش</b></span>
             <i>iTextSharp</i> <u>جهت بررسی فارسی نویسی</u>";
var parsedHtmlElements = HTMLWorker.ParseToList(new StringReader(html), styles);

PdfPCell pdfCell = new PdfPCell { Border = 0 };
pdfCell.RunDirection = PdfWriter.RUN_DIRECTION_RTL;

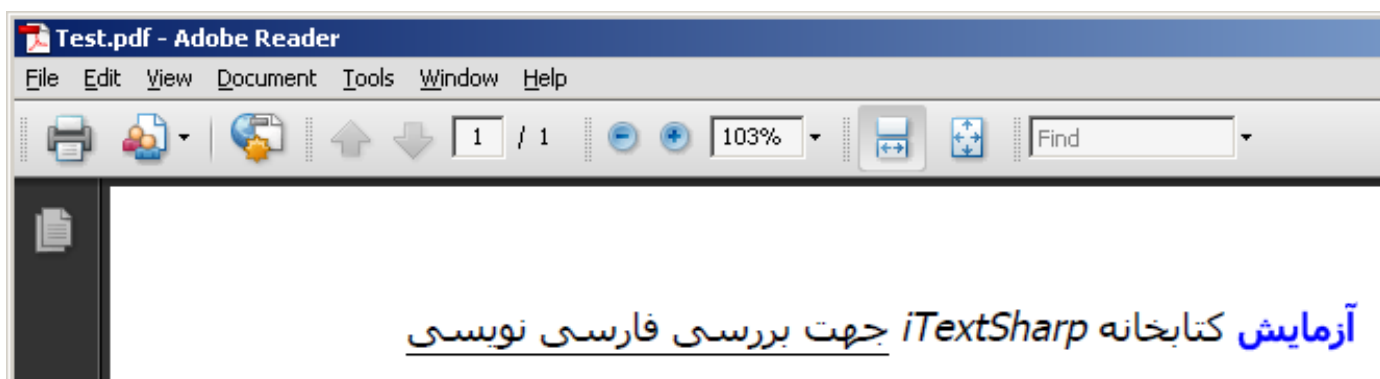
foreach (var htmlElement in parsedHtmlElements)
{
    pdfCell.AddElement(htmlElement);
}

var table1 = new PdfPTable(1);
table1.AddCell(pdfCell);
pdfDoc.Add(table1);
}

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

```

همانطور که ملاحظه می‌کنید ابتدا قلمی در cache قلم‌های این کتابخانه ثبت می‌شود (FontFactory.Register). سپس نوع قلم و encoding آن توسط یک StyleSheet تعریف شده و به HTMLWorker.ParseToList ارسال می‌گردد و در نهایت به کمک یک مان دارای RunDirection، در صفحه نمایش داده می‌شود.



نکته:

ممکن است که به متغیر html ، یک table ساده html را نسبت دهید. در این حالت پس از تنظیم style یاد شده، در هر سلول این html table ، متون فارسی به صورت معکوس نمایش داده خواهند شد که این هم یک نکته‌ی کوچک دیگر دارد:

```
foreach (var htmlElement in parsedHtmlElements)
{
    if (htmlElement is PdfPTable)
    {
        var table = (PdfPTable)htmlElement;
        table.RunDirection = PdfWriter.RUN_DIRECTION_RTL;
        foreach (var row in table.Rows)
        {
            foreach (var cell in row.GetCells())
            {
                cell.RunDirection = PdfWriter.RUN_DIRECTION_RTL;
            }
        }

        pdfCell.AddElement(htmlElement);
    }
}
```

در قسمتی که قرار است المان‌های معادل به pdfCell اضافه شوند، آن‌ها را بررسی کرده و RunDirection آن‌ها را RTL خواهیم کرد.

کاربردها:

بدیهی است این حالت برای تهیه گزارشات پیشرفته‌تر برای مثال تهیه قالب‌هایی که در حین تهیه PDF ، قسمت‌هایی از آن‌ها توسط برنامه نویس Replace می‌شوند، بسیار مناسب است.

همچنین مطلب « [بارگذاری یک یوزر کنترل با استفاده از جی کوئری](#) » و متد RenderUserControl مطرح شده در آن که در نهایت یک قطعه کد HTML را به صورت رشته به ما تحویل می‌دهد، می‌تواند جهت تهیه گزارش‌های پویایی که برای مثال قسمتی از آن یک GridView بایند شده حاصل از یک یوزر کنترل است، مورد استفاده قرار گیرد.

نظرات خوانندگان

نویسنده: A.Karimi
تاریخ: ۱۶:۴۳:۲۶ ۱۳۹۰/۰۶/۱۳

بسیار عالی!

نویسنده: Hossein Hariri
تاریخ: ۱۲:۱۲:۵۸ ۱۳۹۰/۰۶/۲۱

با سلام و تشکر

من از روش شما برای تولید pdf استفاده کردم. ولی متاسفانه align متنهای ایجاد شده Left است و هر کاری میکنم درست نمیشود. لطفا راهنمایی بفرمایید.

نویسنده: وحید نصیری
تاریخ: ۱۵:۲۵:۲۱ ۱۳۹۰/۰۶/۲۱

برای اینکار نیاز است تا style تعریف شده را کمی تغییر داد. به صورت زیر:
(styles.LoadTagStyle(HtmlTags.BODY, HtmlTags.ALIGN, HtmlTags.ALIGN_LEFT

نویسنده: Alisfard
تاریخ: ۲۳:۴۰:۳۸ ۱۳۹۰/۰۶/۲۲

لطفا میشه نمونه کد را برای دانلود قرار دهید چون من هر چی سعی کردم نتونستم اجرا کنم

نویسنده: وحید نصیری
تاریخ: ۲۳:۵۴:۰۷ ۱۳۹۰/۰۶/۲۲

از اینجا قابل دریافت است: [\[^\]](#)

نویسنده: hamidnch2007
تاریخ: ۰۰:۳۵:۰۲ ۱۳۹۰/۰۷/۲۹

من همین روش رو برای گریدویو تودرتو که داخل یه div هستند و آن بصورت runat=server میباشد و آن را با استفاده از متد RenderControl داخل یه String Writer ریختم و اونو بعنوان پارامتر StreamReader پاس دادم. نتیجه این شد که فیلدهای گرید بیرونی درست نمایش داده شدند ولی گریدویو داخلی متنهای آن بصورت برعکس و نچسبیده نمایان شدند ممنون یشتم راهنمایی کنید.

نویسنده: وحید نصیری
تاریخ: ۰۸:۴۰:۳۴ ۱۳۹۰/۰۷/۲۹

parsedHtmlElements تولیدی را دیباگ و سپس نکته آخری رو که در متن بالا عنوان شد باید اعمال کنید.

نویسنده: وحید نصیری
تاریخ: ۱۳:۳۶:۰۶ ۱۳۹۰/۰۷/۲۹

چند نکته عمومی:

- اینجا انجمن نیست. مشکلات عمومی خودتون رو در انجمن‌ها پیگیری کنید.
- اگر خواستید جایی کدی طولانی را ارسال کنید حداقل از سایت <http://pastebin.com> استفاده کنید.
- در مورد این مثال جاری، تا دسترسی به html نهایی تولیدی شما نباشد، دیباگ کردن آن بی‌معنا است.

نویسنده: hamidnch2007
تاریخ: ۱۵:۰۳:۵۰ ۱۳۹۰/۰۷/۲۹

بابت اشتباهم عذرخواهی میکنم.
من فقط میخواستم اگر برایتان امکان پذیر است یه مثال کوچیک از یه html پارس شده که شامل گریدویو تودرتو هست برایم
بزنید و روال را بگویید. جواب شما خیلی کلی بود و بنده متوجه نشدم. باز هم ببخشید.

نویسنده: وحید نصیری
تاریخ: ۲۱:۲۰:۴۶ ۱۳۹۰/۰۷/۲۹

با گرید تو در تو (جداول تو در تو) هم کار می‌کنه؛ فقط اینبار باید اون حلقه‌ای رو که به عنوان نکته گفتم، تبدیل به یک تابع بازگشتی
کنید. به این صورت: <http://pastebin.com/zpMsPmMa>

نویسنده: hamidnch2007
تاریخ: ۲۱:۳۳:۳۳ ۱۳۹۰/۰۷/۲۹

خدا یک در دنیا و هزار در آخرت بهت بده. الهی خیر ببینی. کارت درست. آقای نصیری.
یه سوال دیگه بکنم. اگه من تو گریدویو چک بکس داشته باشم اون رو هم میشه به pdf ارسال کرد؟

نویسنده: وحید نصیری
تاریخ: ۲۲:۰۴:۵۰ ۱۳۹۰/۰۷/۲۹

تعداد تگ‌هایی که iTextSharp ساپورت می‌کنه کم هست. لیست این‌ها رو می‌تونید در کلاس HtmlTags فضای نام
iTextSharp.text.html مشاهده کنید.

یک توصیه کلی:

اگر به دنبال یک راه حل حرفه‌ای برای کارهای پیچیده‌تر HTML to PDF هستید، باید سراغ این نوع کتابخانه‌ها بروید:

[wkhtmltopdf](#), [Convert html to pdf using webkit](#) ([qtwebkit](#))

برای مثال این مورد از WebKit یا همان موتور گوگل کروم استفاده می‌کند. بنابراین HTML parser آن مانند iTextSharp محدود
نیست و فوق العاده حرفه‌ای است.

نویسنده: hamidnch2007
تاریخ: ۲۳:۱۲:۳۷ ۱۳۹۰/۰۷/۲۹

از خدا برای شما بهترین‌ها را آرزو میکنم. امشب مشکل من را حل کردید. امیدوارم که ثمره آن را در زندگی تان ببینی. اگر روزی
بدانم میتوانم برایتان کاری انجام بدم و در توانم باشد دریغ نخواهم کرد. همیشه پیروز باشید.

<> Disqus 2011/10/21

نویسنده: وحید نصیری
تاریخ: ۲۳:۳۹:۰۶ ۱۳۹۰/۰۷/۲۹

ممنون. سلامت باشید.

نویسنده: mkpro
تاریخ: ۱۳:۲۸ ۱۳۹۱/۰۷/۲۵

سلام خسته نباشید

من جدیداً سایتتون رو کشف کردم سایت خیلی مفید و پر باری دارید.

من یه مشکل توی تبدیل html به pdf داشتم وقتی تعداد صفحاتم بیشتر از 2 بشه بهم اختاره Object reference not set to an

instance of an object. و این اختاره مربوط میشه به pdfdoc.add(Table)
ممنون میشم یه راه حلی ارائه بدید.

نویسنده: وحید نصیری
تاریخ: ۱۳۹۱/۰۷/۲۵ ۱۳:۴۰

HTML worker مطرح شده در این مطلب، مدتی است که از طرف نویسندگان آن منسوخ شده اعلام گردیده و دیگری پشتیبانی یا نگهداری نخواهد شد. بنابراین اگر باگی وجود دارد یا هر مطلبی، کسی دیگر به آن رسیدگی نخواهد کرد.
راه حل جایگزین، استفاده از XML Worker است که بجای HTML worker در حال [کار و توسعه می‌باشد](#).

نویسنده: جواد جعفری
تاریخ: ۱۳۹۱/۰۸/۰۴ ۱۴:۸

بابا دمت گرم خیلی خیلی ممنونم واقعا به دردم خورد

نویسنده: مرتضی موسوی
تاریخ: ۱۳۹۲/۰۴/۱۰ ۱۱:۵۸

آقای مهندس , میتونید راهنمایی کنید من چطور میتونم قابلیت Save a Copy as رو غیر فعال کنم . بعبارتی نباید کاربر بتونه از فایل کپی بگیره .

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۴/۱۰ ۱۲:۱۵

مراجعه کنید به مطلب « [رمزنگاری فایل‌های PDF با استفاده از کلید عمومی توسط iTextSharp](#) ». در اینجا توسط مقادیری مانند PdfWriter.ALLOW_COPY و غیره می‌شود روی فایل تولیدی محدودیت ایجاد کرد. ضمنا راه برای برطرف کردن این محدودیت‌ها [هم هست](#).

نویسنده: راد
تاریخ: ۱۳۹۲/۰۸/۰۴ ۱۰:۵۷

باسلام
امکان داره نمونه کد این مثال گذاشته شود
باتشکر

نویسنده: وحید نصیری
تاریخ: ۱۳۹۲/۰۸/۰۴ ۱۳:۲۲

- نمونه کد همان مثالی هست که در متن آورده شده. برای اجرا تنها نیاز به [کتابخانه iTextSharp](#) دارد. (یک برنامه کنسول ساده را ایجاد کنید. کدهای مثال مطلب فوق را در آن paste کنید و بعد ارجاعی را به اسمبلی iTextSharp به آن اضافه نمائید)
- ضمنا افزونه HTMLWorker این کتابخانه منسوخ شده (مطلب جاری) و به [XMLWorker ارتقاء یافته](#).

نویسنده: فیضی
تاریخ: ۱۳۹۲/۱۱/۱۵ ۹:۴۲

سلام.

من کدهای بالا رو استفاده کردم تا محتویات یک فیلد از جدول که محتویاتش از ادیتور TinyMce پر میشه رو به pdf تبدیل کنم.
در ادیتور عکس هم ممکنه درج بشه.

مشکلی که وجود داره اینه که عکسهایی که در ادیتور در سمت راست قرار داده شدن در pdf در سمت چپ قرار می‌گیرن. این مشکل رو چطور میشه رفع کرد؟ نمونه رو می‌تونید از لینک زیر دانلود کنید.

[pdf](#)

ممنون

نویسنده: وحید نصیری
تاریخ: ۹:۵۴ ۱۳۹۲/۱۱/۱۵

- پردازش CSS کتابخانه HTMLWorker خیلی ضعیف و ابتدایی است. به همین جهت آن را کنار گذاشته‌اند و به [XMLWorker](#) کوچ کرده‌اند (HTMLWorker هیچ پشتیبانی رسمی [دیگر ندارد](#)؛ به قسمت `Deprecated. please switch to XML Worker instead` دقت کنید). ضمناً HTMLWorker مشکلات دیگری هم دارد. مثلاً یک تگ `hr` در صفحه باشد، کرش می‌کند. پردازش ویژگی‌های مختلف CSS و HTML تقریباً در آن پیاده سازی نشده و ...

- برای کار با ADO.NET بهتر است این روزها از [Micro ORMs](#) استفاده کنید.

نویسنده: فیضی
تاریخ: ۰:۳ ۱۳۹۲/۱۱/۱۸

ممنون از پاسخ. بله، من `xml worker` ای که شما توضیح داده بودید رو هم تست کرده بودم. `html` ای که شما پاس داده بودید رو به خوبی نشون می‌داد ولی من با همون داده‌های پست قبلی مثال شما رو تست کردم. حروف انگلیسی درست نشون داده میشن ولی فارسی‌ها به شکل نقطه دیده میشن عکسها هم سمت چپ مشاهده می‌شوند

[Sample](#)

یا شاید من اشتباه فهمیدم که `xmlworker` اون قدر قوی هست که عناصر `html` رو می‌تونه با همون استایلی که دارن نشون بده؟ ممنون.

نویسنده: وحید نصیری
تاریخ: ۰:۳۱ ۱۳۹۲/۱۱/۱۸

- در مورد فارسی نویسی در iTextSharp یک دیباگ مرحله به مرحله [قبلاً در سایت](#) مطرح شده. اگر خروجی یونیکد نگرفتید یعنی قلم صحیحی در حال استفاده نیست. کدهایی که [قبلاً ارسال کرده بودم](#) به این نحو است:

```
// روش صحیح تعریف فونت
var systemRoot = Environment.GetEnvironmentVariable("SystemRoot");
FontFactory.Register(Path.Combine(systemRoot, "fonts\\tahoma.ttf"));
```

در کدهای شما به این نحو:

```
var systemRoot = Environment.GetEnvironmentVariable("SystemRoot");
FontFactory.Register(Path.Combine(systemRoot, "c:\\windows\\fonts\\tahoma.ttf"));
```

با توجه به استفاده از `Path.Combine`، مسیری را که معرفی کرده‌اید می‌شود چیزی مانند `c:\windows\c:\windows\fonts\tahoma.ttf`. به همین جهت این فونت یافت نشده و ثبت نمی‌شود (چون دوبار `system root` در آن وجود دارد).

- بله؛ قدرت پردازش CSS در XML Worker آن خیلی بهتر است از HTML Worker.

- در مورد میزان چرخش جدول، `RunDirection = PdfWriter.RUN_DIRECTION_RTL` را با حالت LTR هم تست کنید (`PdfWriter.RUN_DIRECTION_LTR`).