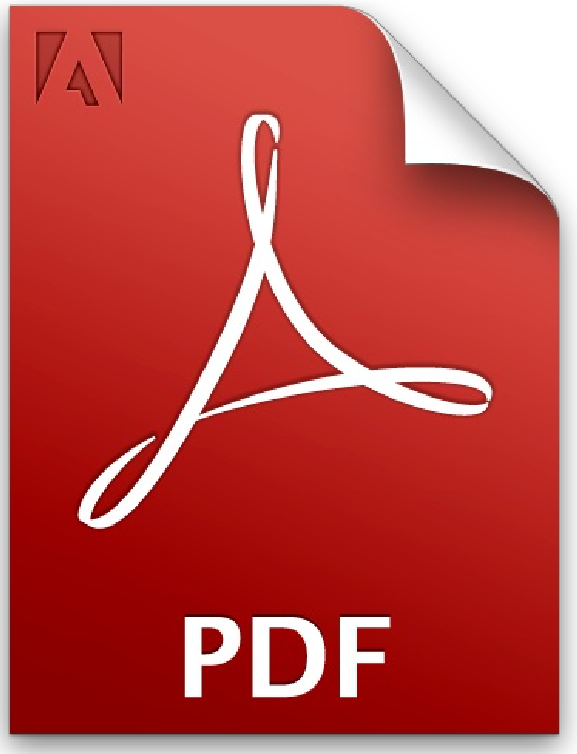
MarkdownToPdf



Convert markdown document to html/pdf using VisualBasic in a super easy way! This is a small VB.NET wrapper utility around `wkhtmltopdf` console tool. You can use it to easily convert Markdown/HTML reports to PDF.

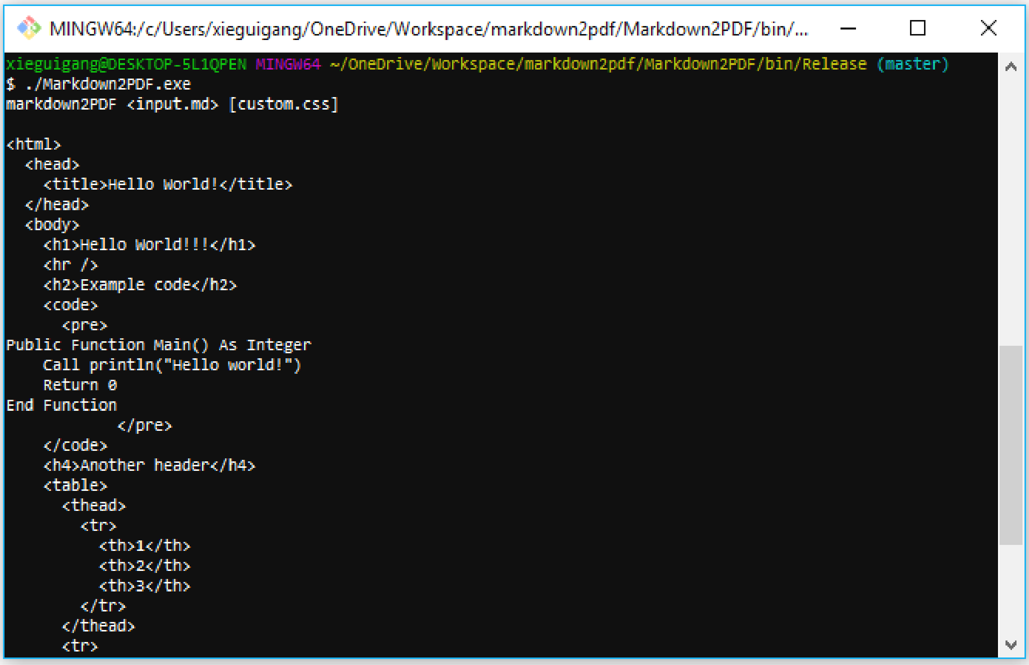
> `wkhtmltopdf` cli wrapper code translate from https://github.com/codaxy/wkhtmltopdf.

Feature

- [+] Convert markdown/html document to PDF
  ☐ Custom CSS style supports. (Works in progress)

How to install

- Clone the source code in this repository
- Open in VisualStudio 2017 and restore the nuget package `sciBASIC#`
- compile
- Extract `wkhtmltopdf` binary distributes from `./dist/wkhtmltopdf.7z` package to app release folder
- Run app in **cmd/bash** or shell scripting for batch converts of the PDF documents.


MINGW64:/c/Users/xieguigang/OneDrive/Workspace/markdown2pdf/Markdown2PDF/bin/...

```
xieguigang@DESKTOP-5L1QPEN MINGW64 ~/OneDrive/Workspace/markdown2pdf/Markdown2PDF/bin/Release (master)
$ ./Markdown2PDF.exe
markdown2PDF <input.md> [custom.css]

<html>
  <head>
    <title>Hello World!</title>
  </head>
  <body>
    <h1>Hello World!!!</h1>
    <hr />
    <h2>Example code</h2>
    <code>
      <pre>
Public Function Main() As Integer
    Call println("Hello world!")
    Return 0
End Function
            </pre>
    </code>
    <h4>Another header</h4>
    <table>
      <thead>
        <tr>
          <th>1</th>
          <th>2</th>
          <th>3</th>
        </tr>
      </thead>
      <tr>
```

Usage

First of all, please make sure the encoding of your console is UTF8. Utf8 encoding is the default encoding of the html document:

```
Console.InputEncoding = Encoding.UTF8
```

Convert: from URL source

```
PdfConvert.Environment.Debug = False
PdfConvert.ConvertHtmlToPdf(New PdfDocument With {
.Url = "input.html"
}, New PdfOutput With {
.OutputFilePath = "output.pdf"
})

PdfConvert.ConvertHtmlToPdf(New PdfDocument With {
.Url = "input.html",
.HeaderLeft = "[title]",
.HeaderRight = "[date] [time]",
.FooterCenter = "Page [page] of [topage]"
}, New PdfOutput With {
.OutputFilePath = "output.pdf"
})
```

Convert: given html content

```
PdfConvert.ConvertHtmlToPdf(New PdfDocument With {
.Html = "<html><h1>test</h1></html>"
}, New PdfOutput With {
.OutputFilePath = "inline.pdf"
})
PdfConvert.ConvertHtmlToPdf(New PdfDocument With {
.Html = "<html><h1>測試</h1></html>"
}, New PdfOutput With {
.OutputFilePath = "inline_cht.pdf"
})
```

Demo: hello world

Actually, you can construct a html document content from `XElement` directly in your VB code:

```
' In this case, we use HTMLDocument object instead of PdfDocument object
Dim html As New HTMLDocument With {
.HTML =
<html>
    <head>
        <title>Hello World!</title>
    </head>
    <body>
        <h1>Hello World!!!</h1>
        <hr/>
        <h2>Example code</h2>
        <code>
            <pre>
                Public Function Main() As Integer
                    Call println("Hello world!")
                    Return 0
                End Function
            </pre>
        </code>
        <h4>Another header</h4>
        <table>
            <thead>
                <tr>
                    <th>1</th>
                    <th>2</th>
                    <th>3</th>
                </tr>
            </thead>
            <tr>
                <td>a</td>
                <td>b</td>
                <td>c</td>
            </tr>
        </table>
        <footer style="position:fixed; font-size:.8em; text-align:right; bottom:0px; margin-left:-25px; height:20px; width:100%;">
            Here is the PDF document footer.
        </footer>
    </body>
</html>
}
Call println(html.GetDocument)
Call PdfConvert.ConvertHtmlToPdf(html, App.HOME & "/hello-world.pdf")
```

CLI Usage

```
# markdown2pdf
markdown2pdf ./input.md
# Due to the reason of markdown parser is compatible with html format,
# so that convert from a html file is also works fine!
markdown2pdf ./input.html
```

Dependence

sciBASIC

```
# http://github.com/xieguigang/sciBASIC
# install via nuget
PM> Install-Package sciBASIC -Pre
```

Licence

This project is available under MIT Licence.