



webpdf 使用教程

HTML 转 PDF 完全指南

版本 1.0

目 录

目录

第一章 简介	1
1.1 什么是 webpdf	1
1.2 主要特性	1
第二章 快速开始	1
2.1 编译项目	2
2.2 运行测试	2
2.3 最简单的示例	2
第三章 文本样式	3
3.1 字体与字号	3
3.1.1 字重与斜体	3
3.1.2 下划线与颜色	3
3.2 文本对齐	3
3.3 行高与段落	3
第四章 表格	4
4.1 基本表格	4
4.2 表格对齐方式	4
4.3 合并单元格	4
4.4 跨页表格	5
第五章 图片与 SVG	6
5.1 位图图片	6
5.2 SVG 矢量图	7
5.3 嵌入 SVG	8
5.4 SVG 渐变	8
5.4.1 线性渐变	8
5.4.2 径向渐变	8
5.5 路径与图形	9
第六章 页面布局	9
6.1 强制分页	9
6.2 满页图片	9
6.3 容器与背景	13
第七章 高级功能	13
7.1 超链接	13
7.2 水平线	14
7.3 页眉与页脚	14

- 7.4 目录生成 14
 - 7.5 封面与扉页 15
- 第八章 API 参考 15
 - 8.1 FPDF 类 15
 - 8.2 完整示例代码 17
- 第九章 常见问题 18
 - 9.1 文字相关 18
 - 9.2 图片相关 18
 - 9.3 分页相关 18
 - 9.4 其他问题 18



第一章 简介

1.1 什么是 webpdf

webpdf 是一个轻量级的 HTML 转 PDF 库，使用 C++ 开发，支持将 HTML 内容渲染为高质量的 PDF 文档。它不需要依赖浏览器内核，而是直接解析 HTML 和 CSS 并生成 PDF 指令，因此启动速度快、内存占用低。

本库特别适合以下场景：

- 服务器端批量生成 PDF 报表和文档
- 嵌入式系统中的 PDF 生成
- 需要精确控制 PDF 输出的应用
- 对启动速度和内存有严格要求的环境

1.2 主要特性

特性	说明
HTML 解析	支持常用 HTML 标签和 CSS 样式
中文支持	完整支持 TrueType 中文字体
SVG 渲染	支持矢量图形和渐变填充
自动分页	智能分页，表格跨页自动重复表头
页眉页脚	支持自定义页眉页脚，可插入页码
目录生成	自动生成带页码的目录，支持点击跳转
满页图片	支持 class="pagefull" 满页渲染

注意：webpdf 不是浏览器内核，而是一个专用的 HTML/CSS 渲染引擎，因此只支持常用的标签和样式。对于复杂的 Web 应用，请使用基于浏览器的方案。



第二章 快速开始

2.1 编译项目

项目使用 Makefile 构建系统，确保系统已安装 g++ 和 make。在项目根目录下执行：

```
cd webpdf
make
```

编译成功后，会生成 `test_html` 可执行文件。

2.2 运行测试

运行测试程序，默认读取 `content.html` 并输出 `output.pdf`：

```
./test_html
```

也可以指定输入和输出文件：

```
./test_html input.html output.pdf
```

2.3 最简单的示例

下面是一个最基本的 HTML 文件，只包含标题和一段文字：

```
<!DOCTYPE html>
<html>
<head>
<meta charset="utf-8">
<title>Hello</title>
<style>
body {
  font-family: msyh;
  font-size: 24px;
}
</style>
</head>
<body>
```



```
<h1>Hello, webpdf!</h1>
<p>这是我的第一个 PDF 文档。</p>
</body>
</html>
```

提示：确保 HTML 文件使用 UTF-8 编码，否则中文可能显示乱码。

第三章 文本样式

3.1 字体与字号

webpdf 支持通过 CSS 设置字体和字号。在使用字体前，必须先通过 C++ API 的 `AddFont()` 方法注册字体文件。

示例：

```
body {
  font-family: "Microsoft YaHei";
  font-size: 24px;
}
```

3.1.1 字重与斜体

可以使用 `<b>` 或 `<strong>` 标签来表示加粗文本，使用 `<i>` 或 `<em>` 标签来表示斜体文本。当然，也可以同时使用 加粗斜体。

3.1.2 下划线与颜色

使用 `<u>` 标签可以添加下划线。使用 `<font color="...">` 可以设置文字颜色，例如 红色、绿色、蓝色。

3.2 文本对齐

这是左对齐的文本。Left aligned text. 中文英文混合显示的效果。

这是居中对齐的文本。Center aligned text.

这是右对齐的文本。Right aligned text.



3.3 行高与段落

行高（line-height）控制每行文字之间的垂直间距。合适的行高可以提高阅读体验。一般来说，行高设置为字号的 1.2 到 1.5 倍比较合适。

这是较小行高的段落。行高为 20px，字号为 14px。文字显得比较紧凑。适合空间有限的场景。但是行高太小会影响阅读速度，眼睛容易疲劳。所以正文内容不建议使用太小的行高。

这是较大行高的段落。行高为 40px，字号为 18px。行间距比较宽松，阅读起来比较舒适。适合长文章阅读，读者可以更轻松地追踪每行的位置。很多技术文档和书籍都采用比较宽松的行高。

第四章 表格

4.1 基本表格

使用 `<table>` 标签创建表格。添加 `border="1"` 属性显示边框。

产品名称	价格	库存
Web 开发框架	¥299	128
移动应用组件库	¥499	56
企业级管理系统	¥1999	12

4.2 表格对齐方式

单元格内容可以设置不同的水平对齐方式和垂直对齐方式：

左对齐	居中对齐	右对齐
文字靠左	文字居中	文字靠右
顶部对齐 第二行 第三行	垂直居中	底部对齐

4.3 合并单元格



使用 `colspan` 合并列，使用 `rowspan` 合并行。

产品信息	价格详情	
软件产品	基础版	¥99
专业版	¥299	
企业版	¥999	
服务项目	技术支持	¥499/年
定制开发	按工计时费	

4.4 跨页表格

当表格内容较多需要跨页显示时，可以添加 `style="position: fixed"`，这样表头会在每一页自动重复显示，方便阅读。

功能模块	基础版	专业版
用户管理	☐ 支持	☐ 支持
角色权限	☐ 基础	☐ 完整
数据报表	☐ 基础报表	☐ 高级报表
导出功能	☐ Excel	☐ Excel/PDF/Word
消息通知	☐ 不支持	☐ 支持
工作流引擎	☐ 不支持	☐ 支持
API 接口	☐ 基础接口	☐ 完整接口
第三方集成	☐ 不支持	☐ 支持
数据备份	☐ 手动备份	☐ 自动备份
操作日志	☐ 基础日志	☐ 完整审计
多语言支持	☐ 不支持	☐ 支持

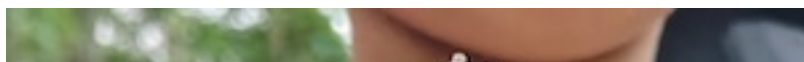


功能模块	基础版	专业版
移动端适配	❑ 不支持	❑ 支持
技术支持	社区支持	7x24 专人支持
升级服务	小版本升级	全版本升级
定制开发	❑ 不支持	可选服务
部署方式	公有云	公有云/私有部署
SLA 保障	99%	99.9%
数据迁移服务	❑ 不支持	❑ 支持
培训服务	在线文档	现场培训
安全认证	基础	等保三级
性能优化	标准	深度优化
插件市场	❑ 不支持	❑ 支持
开放平台	❑ 不支持	❑ 支持
多租户	❑ 不支持	❑ 支持
BI 分析	❑ 不支持	❑ 支持
AI 助手	❑ 不支持	❑ 可选
价格（每年）	¥2,999	¥9,999
合计功能数	12 项	28 项

第五章 图片与 SVG

5.1 位图图片

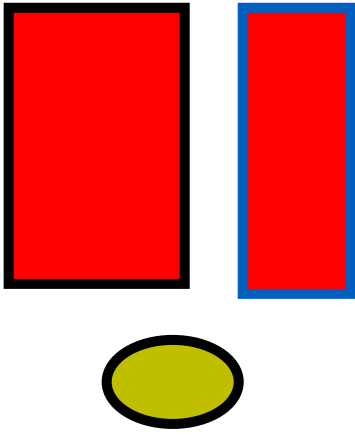
webpdf 支持 PNG、JPG 等常见位图格式。使用 `<img>` 标签插入图片：





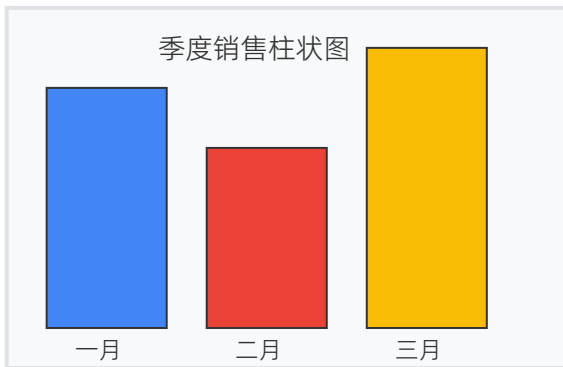
5.2 SVG 矢量图

SVG（可缩放矢量图形）可以无损缩放，适合图表和图标。可以通过 `<img>` 引用 SVG 文件，也可以直接在 HTML 中嵌入 SVG 代码。



5.3 嵌入 SVG

在 HTML 中直接嵌入 SVG 代码，可以获得更灵活的控制：



5.4 SVG 渐变

webpdf 支持线性渐变和径向渐变两种 SVG 渐变效果。

5.4.1 线性渐变

线性渐变沿着一条直线方向逐渐变化颜色：



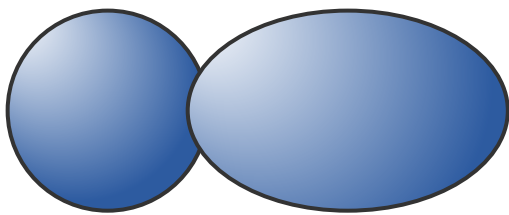
垂直方向的线性渐变：



5.4.2 径向渐变

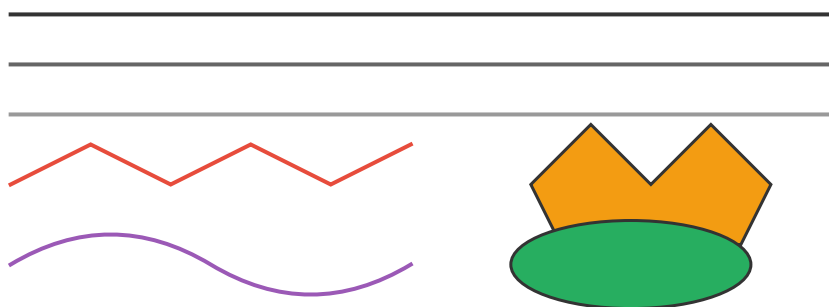


径向渐变从中心点向外扩散，适合绘制球体和光晕效果：



5.5 路径与图形

SVG 支持多种基本图形和路径绘制：



第六章 页面布局

6.1 强制分页

有时需要在特定位置强制分页，可以使用 `page-break-before: always` 样式：

```
<div style="page-break-before: always;"></div>
```

上面这个空的 div 不会产生任何可见内容，只会触发一次分页。

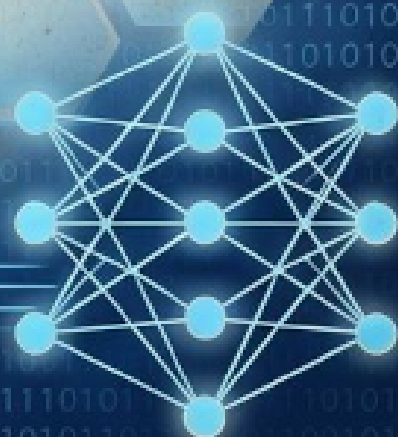
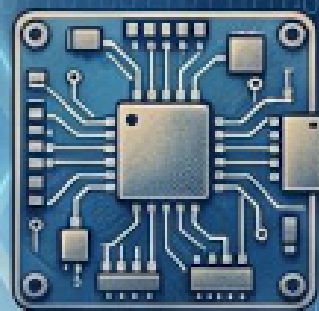
6.2 满页图片

使用 `class="pagefull"` 可以让图片占满整个页面，不显示页眉、页脚和边距。图片会按比例缩放并居中显示。

```
<p class="pagefull"></p>
```

下面是一张满页的 PNG 图片：

人工智能 大开发





上面这张图片占满了一整页，没有页眉和页脚。现在是新的一页，页眉页脚恢复正常显示。

SVG 矢量图也可以满页显示：





这又是新的一页，页眉页脚正常显示。满页图片功能非常适合封面、插图、图表等需要整页展示的内容。

小技巧：pagefull 类可以加在任何容器元素上，容器内的第一个 img 元素会被用于满页渲染，容器本身和其他子元素不会显示。

6.3 容器与背景

使用 `<div>` 可以创建块级容器，设置背景色、边框、圆角等样式：

信息提示框

这是一个蓝色主题的提示框。可以用来显示重要的信息、注意事项等。

警告提示

这是一个警告提示框，左侧有醒目的红色边框。

HOT

热门推荐

带背景图和圆角边框的容器样式。

fieldset提示框

这是一个蓝色主题的提示框。可以用来显示重要的信息、注意事项等。

第七章 高级功能

7.1 超链接

使用 `<a>` 标签可以创建超链接，在 PDF 中点击可以跳转到指定网页。例如：访问[示](#)



例[网站](#)了解更多信息。

长链接也能正确换行显示：<https://www.example.com/very/long/path/to/document.html?query=value¶m=test&foo=bar&baz=qux>

7.2 水平线

使用 `<hr>` 标签可以插入水平线，用于分隔内容：

默认样式的水平线：



红色粗线：



绿色虚线：



蓝色点线，50% 宽度居中：



橙色短线条，右对齐：



7.3 页眉与页脚

使用 `class="pageheader"` 和 `class="pagefooter"` 可以自定义每一页的页眉和页脚。本文档的页眉和页脚就是使用这个功能实现的。

页眉页脚中可以使用两个特殊占位符：

占位符	说明
{pagenum}	当前页码
{pagetotal}	文档总页数

注意：封面页和目录页默认不显示页眉页脚。满页图片（pagefull）的页面也不显示页眉页脚。



7.4 目录生成

webpdf 可以自动生成目录。给标题添加 `class="catalogue1"`、`catalogue2`、`catalogue3` 即可自动加入目录。

类名	目录级别	对应标题
catalogue1	一级目录	h1 ~ h3
catalogue2	二级目录	h4 ~ h5
catalogue3	三级目录	h5 ~ h6

生成的目录带有页码，并且可以点击跳转到对应章节。

7.5 封面与扉页

使用 `class="pagecover"` 可以创建封面页，封面页不显示页眉页脚，内容从页面顶部开始。

使用 `class="pagecatalogue"` 可以创建目录页背景，配合 `class="cataloguecontent"` 自动生成目录内容。

第八章 API 参考

8.1 FPDF 类

FPDF 是 PDF 文档的核心类，用于创建和操作 PDF 文件。

方法	说明
<code>FPDF()</code>	构造函数，创建一个新的 PDF 文档
<code>AddPage()</code>	添加一页新页面
<code>AddFont(family, style, file, dir)</code>	注册字体文件
<code>SetFont(family, style, size)</code>	设置当前字体



方法	说明
<code>WriteHTML(html, line_height)</code>	渲染 HTML 内容到 PDF
<code>Output(dest, name)</code>	输出 PDF 文件
<code>SetShowHeader(show)</code>	设置是否显示页眉
<code>SetShowFooter(show)</code>	设置是否显示页脚
<code>SetShowHeaderFooter(show)</code>	同时设置页眉页脚显示
<code>PageNo()</code>	获取当前页码
<code>SetPageNumberOffset(offset)</code>	设置页码偏移量
<code>SetLeftMargin(margin)</code>	设置左边距
<code>SetRightMargin(margin)</code>	设置右边距
<code>SetTopMargin(margin)</code>	设置上边距
<code>SetAutoPageBreak(auto, margin)</code>	设置自动分页
<code>GetPageWidth()</code>	获取页面宽度
<code>GetPageHeight()</code>	获取页面高度
<code>GetX() / GetY()</code>	获取当前坐标
<code>SetXY(x, y)</code>	设置当前坐标
<code>Image(file, x, y, w, h)</code>	插入图片
<code>Rect(x, y, w, h, style)</code>	绘制矩形
<code>SetFillColor(r, g, b)</code>	设置填充颜色
<code>SetDrawColor(r, g, b)</code>	设置描边颜色



方法	说明
<code>SetTextColor(r, g, b)</code>	设置文字颜色
<code>SetLineWidth(width)</code>	设置线宽
<code>AddAxialShading(...)</code>	添加轴向渐变
<code>ShadeFill(x, y, w, h, id)</code>	使用渐变填充矩形

8.2 完整示例代码

```
#include <iostream>
#include <fstream>
#include <sstream>
#include <string>
#include "fpdf.h"

std::string read_file(const std::string& path) {
    std::ifstream f(path);
    if (!f.is_open())
        throw std::runtime_error("Cannot open file: " + path);
    std::stringstream ss;
    ss << f.rdbuf();
    return ss.str();
}

int main(int argc, char* argv[]) {
    try {
        std::string html_file = "content.html";
        std::string pdf_file = "output.pdf";

        if (argc > 1) html_file = argv[1];
        if (argc > 2) pdf_file = argv[2];

        std::string html = read_file(html_file);

        webpdf::FPDF pdf;
        pdf.AddPage();
        pdf.AddFont("msyh", "", "msyh.ttf", "font/");
        pdf.SetFont("msyh", "", 14);
        pdf.WriteHTML(html);
        pdf.Output("F", pdf_file);

        std::cout << "PDF generated: " << pdf_file << std::endl;
        return 0;
    }
```



```
} catch (const std::exception& e) {  
    std::cerr << "Error: " << e.what() << std::endl;  
    return 1;  
}  
}
```

第九章 常见问题

9.1 文字相关

问题	解决方案
中文显示乱码或方框	确保已使用 AddFont() 注册中文字体，且 CSS 中 font-family 名称匹配
字体粗细不起作用	需要注册对应字重的字体文件，webpdf 不会自动模拟加粗
文字换行不正确	检查是否有特殊字符或空格，中文按字符宽度换行

9.2 图片相关

问题	解决方案
图片不显示	检查图片路径是否正确，路径相对于程序运行目录
SVG 显示为黑色	检查 SVG 语法是否正确，渐变需要在 defs 中定义
SVG 渐变显示不正确	确保渐变 ID 正确，使用 url(#id) 格式引用

9.3 分页相关

问题	解决方案
pagebreak 前有空白页	使用空 div + page-break-before: always，不要在里面放内容
表格跨页表头不重复	添加 style="position: fixed" 到 table 标签
pagefull 图片有页眉页脚	确保 class="pagefull" 加在容器上，且包含 img 子元素



9.4 其他问题

问题	解决方案
目录页码不对	目录页码是估算的，复杂布局可能有偏差
HTML 中某些样式不生效	webpdf 只支持常用 CSS 属性，查看教程中的支持列表
生成的 PDF 文件很大	使用压缩的图片，减少大尺寸位图的使用

写在最后：webpdf 是一个持续发展的项目，托管于 <https://github.com/hggq/webpdf>。如果你遇到了问题或需要新功能，欢迎反馈和贡献代码。希望本教程能帮助你快速上手，制作出精美的 PDF 文档！