# MCP Document Converter

一个基于 Model Context Protocol (MCP) 的文档转换工具，支持多种文档格式之间的转换。

## 🚀 特性

- 📄 支持多种文档格式：PDF、Word (DOCX)、Markdown、HTML、TXT

- 🔄 灵活的格式转换矩阵

- 🖼️ 图片提取功能

- 📊 文档信息分析

- 🔧 CLI 和 MCP 服务器双模式

- 📦 支持批量转换

- 🎯 通过 npx 直接使用，无需安装

## 📋 支持的格式

### 输入格式

- PDF (.pdf)

- Microsoft Word (.docx, .doc)

- HTML (.html, .htm)

- Markdown (.md, .markdown)

- 纯文本 (.txt)

### 输出格式

- PDF (.pdf)

- Microsoft Word (.docx)

- HTML (.html)

- Markdown (.md)

- 纯文本 (.txt)

### 转换矩阵

- \*\*PDF\*\* → TXT, MD, HTML

- \*\*DOCX\*\* → TXT, MD, HTML, PDF

- \*\*HTML\*\* → TXT, MD, PDF

- \*\*MD\*\* → HTML, PDF, TXT

- \*\*TXT\*\* → HTML, MD, PDF

## 🛠️ 安装

### 通过 npx 使用（推荐）

```bash

npx mcp-document-converter --help

```

### 全局安装

```bash

npm install -g mcp-document-converter

```

### 本地开发

```bash

git clone <repository-url>

cd mcp-document-converter

npm install

npm run build

```

## 📖 使用方法

### CLI 模式

#### 转换单个文件

```bash

# 将 PDF 转换为 Markdown

npx mcp-document-converter convert -i document.pdf -o output.md -f md

# 将 Word 文档转换为 HTML

npx mcp-document-converter convert -i report.docx -o report.html -f html

# 将 Markdown 转换为 PDF

npx mcp-document-converter convert -i readme.md -o readme.pdf -f pdf

```

#### 获取文档信息

```bash

npx mcp-document-converter info -f document.pdf

```

#### 批量转换

```bash

# 将目录中的所有文档转换为 HTML

npx mcp-document-converter batch -d ./documents -o ./output -f html

# 保持原始格式

npx mcp-document-converter batch -d ./docs -o ./converted --preserve-formatting -f md

```

#### 查看支持的格式

```bash

npx mcp-document-converter formats

```

#### 查看使用示例

```bash

npx mcp-document-converter examples

```

### MCP 服务器模式

启动 MCP 服务器：

```bash

npx mcp-document-converter mcp

```

#### MCP 工具

1. \*\*convert\_document\*\* - 转换文档格式

```json

{

"input\_path": "/path/to/input.pdf",

"output\_path": "/path/to/output.md",

"target\_format": "md",

"options": {

"preserve\_formatting": true,

"extract\_images": false

}

}

```

2. \*\*get\_document\_info\*\* - 获取文档信息

```json

{

"file\_path": "/path/to/document.pdf"

}

```

3. \*\*list\_supported\_formats\*\* - 列出支持的格式

```json

{}

```

## ⚙️ 配置选项

### 转换选项

- `preserve\_formatting`: 保持原始格式（默认：true）

- `extract\_images`: 提取图片到单独文件（默认：false）

- `image\_output\_dir`: 图片输出目录

### CLI 选项

- `-i, --input <path>`: 输入文件路径

- `-o, --output <path>`: 输出文件路径

- `-f, --format <format>`: 目标格式

- `--preserve-formatting`: 保持原始格式

- `--extract-images`: 提取图片

- `--image-dir <path>`: 图片保存目录

## 🔧 开发

### 项目结构

```

mcp-document-converter/

├── src/

│ ├── index.ts # MCP 服务器入口

│ ├── converter.ts # 文档转换核心逻辑

│ └── cli.ts # CLI 入口

├── dist/ # 编译输出

├── package.json

├── tsconfig.json

└── README.md

```

### 构建

```bash

npm run build

```

### 开发模式

```bash

npm run dev

```

### 测试

```bash

# 测试 CLI

node dist/cli.js convert -i test.pdf -o test.md -f md

# 测试 MCP 服务器

node dist/index.js

```

## 📦 依赖

### 核心依赖

- `@modelcontextprotocol/sdk`: MCP SDK

- `mammoth`: Word 文档处理

- `pdf-parse`: PDF 解析

- `turndown`: HTML 到 Markdown 转换

- `marked`: Markdown 到 HTML 转换

- `puppeteer`: PDF 生成

- `commander`: CLI 框架

### 系统要求

- Node.js >= 16.0.0

- 对于 PDF 生成功能，需要 Chromium（通过 Puppeteer 自动安装）

## 🐛 故障排除

### 常见问题

1. \*\*Puppeteer 安装失败\*\*

```bash

# 设置 Puppeteer 下载镜像

npm config set puppeteer\_download\_host=https://npm.taobao.org/mirrors

npm install

```

2. \*\*PDF 转换失败\*\*

- 确保系统有足够内存

- 检查输入 PDF 文件是否损坏

- 尝试使用 `--preserve-formatting=false` 选项

3. \*\*Word 文档转换问题\*\*

- 仅支持 .docx 格式（不支持旧的 .doc 格式的完整功能）

- 复杂的格式可能无法完全保留

4. \*\*权限问题\*\*

- 确保对输入文件有读取权限

- 确保对输出目录有写入权限

## 📄 许可证

MIT License

## 🤝 贡献

欢迎提交 Issue 和 Pull Request！

## 📞 支持

如果遇到问题，请：

1. 查看本 README 的故障排除部分

2. 在 GitHub 上提交 Issue

3. 提供详细的错误信息和复现步骤

---

\*\*享受文档转换的便利！\*\* 🎉