

# MrxDown PDF Export - Qualitätstest

---

Dieses Dokument demonstriert alle PDF-Export-Verbesserungen in Version 0.2.1.

---

## 1. Paragraph-Handling & Leerzeilen

---

Dies ist ein normaler Paragraph mit optimiertem Spacing. Der Text verwendet eine professionelle Zeilenhöhe von 1.7 und automatische Silbentrennung.

Dies ist ein zweiter Paragraph. Beachten Sie den konsistenten Abstand (0.8em) zwischen den Absätzen.

Hier ein Paragraph mit einem erzwungenen Zeilenumbruch innerhalb des Paragraphs.

Ein weiterer Paragraph nach dem Umbruch.

---

## 2. Emoji-Unterstützung 🎉

---

Emojis werden jetzt korrekt dargestellt:

- 📄 Dokumente und Text
- 🚀 Rakete für Batch-Export
- ✅ Checkmarks funktionieren
- 🎨 Kreative Icons
- 💡 Ideen-Glühbirne
- 🔍 Such-Lupe
- ⚡ Performance-Blitz
- ⭐ Sterne für Highlights

**Kombination:** Text mit Emoji 🇩🇪 mitten im Satz funktioniert perfekt!

---

## 3. Hyperlinks mit URL-Anzeige

---

Hier sind verschiedene Link-Typen:

- [Externe Website](https://github.com/pepperonas/mrxdn) (<https://github.com/pepperonas/mrxdn>).
- [Dokumentation](https://docs.github.com) (<https://docs.github.com>).
- [Anchor-Link](#) (URL wird nicht angezeigt)
- Inline-Link: Besuchen Sie [unsere Homepage](https://pepperonas.com) (<https://pepperonas.com>) für mehr Info.

**Erwartet:** URLs erscheinen nach dem Link in Klammern und grauer Farbe.

---

## 4. Listen-Formatierung

---

### Ungeordnete Liste

- Erster Punkt mit optimiertem Abstand
- Zweiter Punkt
  - Verschachtelter Unterpunkt (Ebene 2)
  - Weiterer Unterpunkt
    - Noch tiefer verschachtelt (Ebene 3)
- Zurück zur Hauptebene

### Geordnete Liste

1. Erste Position
2. Zweite Position
  1. Sub-Position 2.1
  2. Sub-Position 2.2
3. Dritte Position

### Task-Liste

- ☐ Aufgabe nicht erledigt

- ☒ Aufgabe erledigt
  - ☐ Weitere offene Aufgabe
  - ☒ Abgeschlossen
- 

## 5. Code-Blocks & Inline-Code

---

### Inline-Code

Verwenden Sie `const variable = 'value';` für JavaScript-Variablen. Die Funktion `Array.map()` ist sehr nützlich.

### JavaScript-Code-Block

```
function exportToPDF(content) {  
  const pdfData = await printToPDF({  
    pageSize: 'A4',  
    printBackground: true,  
    preferCSSPageSize: true  
  });  
  
  return pdfData;  
}
```

### Python-Code-Block

```
def calculate_sum(numbers):  
    """Berechnet die Summe einer Liste."""  
    total = sum(numbers)  
    return total  
  
result = calculate_sum([1, 2, 3, 4, 5])  
print(f"Summe: {result}")
```

## Shell-Code

```
npm install
npm run build-mac
npm start
```

## 6. Tabellen mit Page-Break-Control

### Einfache Tabelle

Feature	Status	Priorität
Leerzeilen-Handling	✅ Implementiert	Hoch
Emoji-Support	✅ Implementiert	Hoch
Hyperlinks	✅ Implementiert	Hoch
Listen	✅ Implementiert	Mittel
Seitenumbruch	✅ Implementiert	Hoch

### Erweiterte Tabelle

Tool	Sprache	Version	License	Verwendung
Electron	JavaScript	28.0.0	MIT	App Framework
Marked	JavaScript	12.0.1	MIT	Markdown Parser
DOMPurify	JavaScript	3.0.9	Apache-2.0	Sanitization
Node.js	JavaScript	20.x	MIT	Runtime

**Hinweis:** Table-Header wiederholt sich automatisch auf neuen Seiten.

---

## 7. Blockquotes

---

*Dies ist ein einfaches Blockquote mit optimiertem Styling. Es umfasst mehrere Zeilen und hat eine professionelle Darstellung.*

**Wichtiger Hinweis:** Blockquotes bleiben zusammen und werden nicht über Seiten hinweg getrennt (page-break-inside: avoid).

---

## 8. Überschriften-Hierarchie

---

# Hauptüberschrift (H1)

---

Dies sollte die einzige H1 im Dokument sein.

## Kapitel-Überschrift (H2)

---

Wichtige Sections verwenden H2.

### Unterkapitel (H3)

Details in H3-Überschriften.

#### Detail-Section (H4)

Noch detailliertere Informationen.

#### Feinste Details (H5)

Sehr spezifische Inhalte.

## Minimal-Ebene (H6)

Die kleinste Überschriften-Ebene.

**Page-Break-Control:** Überschriften bleiben mit dem folgenden Inhalt zusammen!

---

# 9. Typografie-Features

---

## Text-Formatierung

- **Fettgedruckter Text** für Betonung
- *Kursiver Text* für Hervorhebungen
- ***Fett und kursiv*** kombiniert
- ~~Durchgestrichener Text~~

## Sonderzeichen & Symbole

- Copyright: ©
- Trademark: ™
- Registered: ®
- Währung: € \$ £ ¥
- Mathematik:  $\approx \neq \leq \geq \infty$
- Pfeile: → ← ↑ ↓ ↔
- Sterne: ★ ☆ ◆

## Widows & Orphans

Dieser Paragraph demonstriert die Widow/Orphan-Kontrolle. Der Text ist lang genug, um über mehrere Zeilen zu gehen und zeigt, dass mindestens 3 Zeilen am Anfang oder Ende einer Seite bleiben. Dies verhindert unschöne einzelne Zeilen am Seitenbeginn oder -ende.

---

## 10. Horizontale Trennlinien

---

Trennlinien verwenden optimiertes Styling:

---

Zwischen zwei Sections.

---

## 11. Verschachtelte Strukturen

---

### Listen in Blockquotes

#### ***Wichtige Features:***

- 1. PDF-Export mit A4-Format*
- 2. Automatische Seitenumbrüche*
- 3. Professionelle Typografie*
  - Optimierte Zeilenhöhe*
  - Widows/Orphans-Kontrolle*
  - Justierter Text*

### Code in Listen

#### 1. Installation

```
git clone https://github.com/pepperonas/mrxdown.git
cd mrxdown
npm install
```

## 2. Entwicklung

```
npm start
```

## 3. Build

```
npm run build-mac
```

---

# 12. Lange Paragraphen für Seitenumbruch-Test

---

Lorem ipsum dolor sit amet, consectetur adipiscing elit. Sed do eiusmod tempor incididunt ut labore et dolore magna aliqua. Ut enim ad minim veniam, quis nostrud exercitation ullamco laboris nisi ut aliquip ex ea commodo consequat.

Duis aute irure dolor in reprehenderit in voluptate velit esse cillum dolore eu fugiat nulla pariatur. Excepteur sint occaecat cupidatat non proident, sunt in culpa qui officia deserunt mollit anim id est laborum.

Sed ut perspiciatis unde omnis iste natus error sit voluptatem accusantium doloremque laudantium, totam rem aperiam, eaque ipsa quae ab illo inventore veritatis et quasi architecto beatae vitae dicta sunt explicabo.

Nemo enim ipsam voluptatem quia voluptas sit aspernatur aut odit aut fugit, sed quia consequuntur magni dolores eos qui ratione voluptatem sequi nesciunt. Neque porro quisquam est, qui dolorem ipsum quia dolor sit amet, consectetur, adipisci velit.

---

# 13. Performance-Informationen

---

## CSS-Größe

- Vorher: ~1.2 KB
- Nachher: ~3.8 KB
- **Zuwachs:** +2.6 KB für bessere Qualität



## Rendering-Zeit

- Vorher: 1000ms
  - Nachher: 1500ms
  - **Grund:** Komplexere Styles für professionelles Ergebnis
- 

## Zusammenfassung

---

Dieser Test-Export demonstriert:

✅ **9 von 10** Hauptverbesserungen implementiert ⌚ **1 Feature** für v0.3.0 geplant (PDF-Metadaten)

## Implementiert

1. Optimiertes Paragraph-Handling
2. Emoji-Support
3. Hyperlinks mit URLs
4. Listen-Formatierung
5. Seitenumbruch-Kontrolle
6. Bild-Optimierung
7. Code-Block-Enhancement
8. Tabellen-Verbesserungen
9. Professionelle Typografie

## Geplant (v0.3.0)

10. PDF-Metadaten (Titel, Autor, Keywords)
- 

Entwickelt mit ❤️ von Martin Pfeffer © 2025 MrxDown Version 0.2.1