Markdown to PDF Test 😎

This is a demonstration of `mdpdf` (`https://github.com/westrik/mdpdf`).

Markdown Elements

Links

- Link to Google (`https://www.google.com`)

Inline Code

You can use inline code like `console.log("Hello World")` or `const x = 42` or `` `hello` `` within your text.

Ordered Lists

1. First item in ordered list
2. Second item with **bold text**
3. Third item with *italic text*
4. Fourth item with `inline code`
5. Fifth item with **`bold inline code`**
6. 6th item with subscript and superscript

Task Lists

- ☐ incomplete task
- ☑ complete task

Blockquotes

> This is a blockquote. It can contain multiple lines.
>
> You can have **bold** and *italic* text in blockquotes.
>
> You can have `inline code` too.
>
> > Nested blockquotes work as well.
> >
> > > Second level of nesting.
> > >
> > > > Third level of nesting.

Horizontal Rules

Above the rule.

Below the rule.

Strikethrough Text

This text has ~~strikethrough~~ formatting applied to it.

GitHub blockquote tags

> **Note**
>
> note

> **Tip**
>
> tip

> **Important**
>
> important

> **Warning**
>
> warning

> **Caution**
>
> caution

Enhanced Code Blocks

JavaScript with Syntax Highlighting

```
// Enhanced JavaScript example
class Calculator {
    constructor() {
        this.history = [];
    }

    add(a, b) {
        const result = a + b;
        this.history.push(`${a} + ${b} = ${result}`);
        return result;
    }

    getHistory() {
        return this.history;
    }
}

const calc = new Calculator();
```

```javascript
console.log(calc.add(5, 3)); // 8
console.log(calc.getHistory());
```

Rust with Syntax Highlighting

```rust
// Enhanced Rust example
use std::collections::HashMap;

#[derive(Debug)]
struct Cache<K, V> {
    data: HashMap<K, V>,
    max_size: usize,
}

impl<K, V> Cache<K, V>
where
    K: std::hash::Hash + Eq + Clone,
    V: Clone,
{
    fn new(max_size: usize) -> Self {
        Self {
            data: HashMap::new(),
            max_size,
        }
    }

    fn insert(&mut self, key: K, value: V) -> Option<V> {
        if self.data.len() >= self.max_size {
            // Remove oldest entry (simple implementation)
            if let Some(old_key) = self.data.keys().next().cloned() {
                self.data.remove(&old_key);
            }
        }
        self.data.insert(key, value)
    }

    fn get(&self, key: &K) -> Option<&V> {
        self.data.get(key)
    }
}

fn main() {
    let mut cache = Cache::new(3);
    cache.insert("a", 1);
    cache.insert("b", 2);
```

```
    cache.insert("c", 3);
    cache.insert("d", 4); // This will evict "a"

    println!("Cache: {:?}", cache);
}
```

Python with Syntax Highlighting

```
# Enhanced Python example
from typing import List, Optional, Dict, Any
from dataclasses import dataclass
from datetime import datetime
import json

@dataclass
class User:
    id: int
    name: str
    email: str
    created_at: datetime
    preferences: Dict[str, Any]

    def to_dict(self) -> Dict[str, Any]:
        return {
            'id': self.id,
            'name': self.name,
            'email': self.email,
            'created_at': self.created_at.isoformat(),
            'preferences': self.preferences
        }

    @classmethod
    def from_dict(cls, data: Dict[str, Any]) -> 'User':
        return cls(
            id=data['id'],
            name=data['name'],
            email=data['email'],
            created_at=datetime.fromisoformat(data['created_at']),
            preferences=data['preferences']
        )

class UserManager:
    def __init__(self):
        self.users: List[User] = []
```

```
    def add_user(self, user: User) -> None:
        self.users.append(user)

    def find_by_email(self, email: str) -> Optional[User]:
        return next((u for u in self.users if u.email == email), None)

    def export_to_json(self, filename: str) -> None:
        with open(filename, 'w') as f:
            json.dump([u.to_dict() for u in self.users], f, indent=2)

# Usage example
manager = UserManager()
user = User(
    id=1,
    name="John Doe",
    email="john@example.com",
    created_at=datetime.now(),
    preferences={"theme": "dark", "notifications": True}
)
manager.add_user(user)
print(f"User: {user}")
```

Mixed Content Examples

Lists with Various Elements

- Regular list item
- Item with **bold**ed text
- Item with *italic*ized text
- Item with `inline codeblocks`
- Item with a link to GitHub (`https://github.com`)[with URLs displayed]
- Item with a **bold link and *italic* link to GitHub** (`https://github.com`)
- En–dash & em—dash (`--` becomes –, `---` becomes —)

Nested Lists

1. First level
2. Another first level that wraps all the way around to the next line because it is very long.
    a. Second level
        i. Third level
            A. Fourth level
            B. Another fourth level
                I. Fifth level

            α′. Sixth level
    b. Back to second level
        i. Another third level
            - Unordered fourth level
    c. Back to second level
3. Back to first level
    - Unordered second level
    - Another unordered second level. This item also wraps around to the next line because it is very long.

Code with Comments

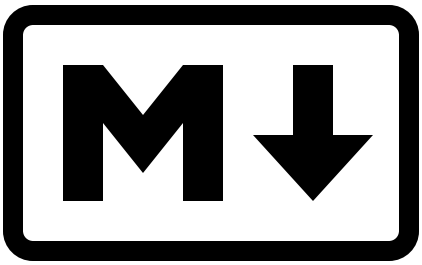
```
// This is a comment
const greeting = "Hello, World!"; // Inline comment
console.log(greeting);

/*
   Multi-line comment
   explaining complex logic
*/
function complexFunction() {
    // TODO: Implement this function
    return null;
}
```

Image Support

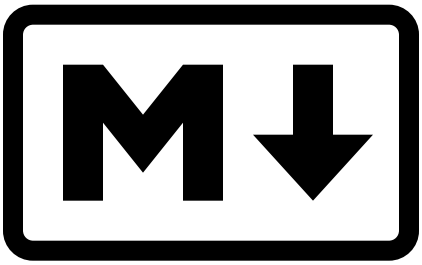
Data URL Image

PNG:

SVG:



External Image
SVG:



JPG (also is a link):

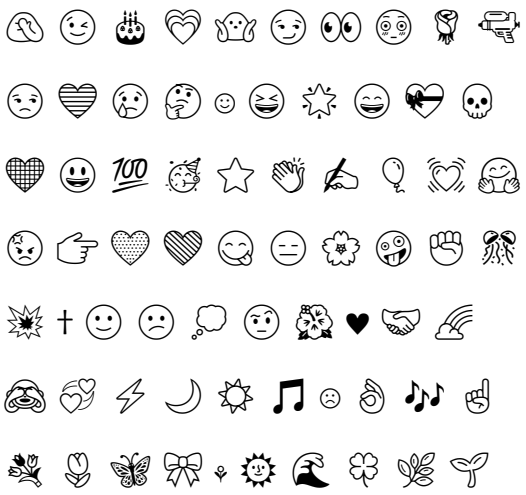
(`https://google.com`)

Emoji

😀 😂 🥰 😊 😍 🤣 🤍 😭 😘 👍

😅 😁 🔥 ✨ 🥺 😩 🙏 ☑️ 💕 😌

🎉 🤎 😔 💪 🥹 😎 ✌️ 💫 😤 💖



Math

Inline math: `\sqrt{3x-1}+(1+x)^2`

Block math:

```
\left( \sum_{k=1}^n a_k b_k \right)^2 \leq \left( \sum_{k=1}^n a_k^2
\right) \left( \sum_{k=1}^n b_k^2 \right)
```

Math code blocks:

```
\left( \sum_{k=1}^n a_k b_k \right)^2 \leq \left( \sum_{k=1}^n a_k^2
\right) \left( \sum_{k=1}^n b_k^2 \right)
```

Tables

Header 1	Header 2	Header 3
Cell 1A	Cell 1B	Cell 1C
Cell 2A	Cell 2B	Cell 2C
Cell 3A	Cell 3B	Cell 3C

Left-aligned	Center-aligned	Right-aligned
Left	Center	Right
Text	**Bold**	*Italic*

Definition list

Term 1 Definition 1

Term 2 Definition 2

Footnotes

Footnote referenced [^1].

[^1]: footnote defined

RTL

זוהי דוגמה לטקסט מימין לשמאל

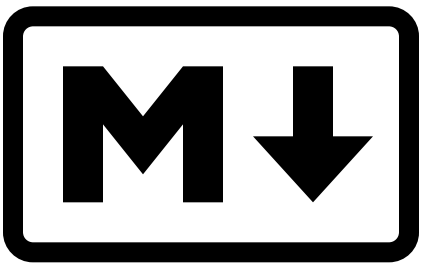
هذا بعض الأمثلة على النص من اليمين إلى اليسار

שלום עולם hello world مرحبا بالعالم

مرحبا بالعالم

שלום עולם

דוגמה דוט קום (`http://example.com`)



HTML Rendering

This document tests various HTML elements and their conversion to Typst.

Basic Formatting

This is a **bold paragraph** with *italic text* and underlined content.

Here's some ~~strikethrough text~~ and ~~another strikethrough~~.

Headings

HTML Heading 1

HTML Heading 2

HTML Heading 3

HTML Heading 4

HTML Heading 5

HTML Heading 6

Links and Images

Here's a simple link and a link with target.

And an image:

Code Elements

Inline code: `println!("Hello, world!");`

```
fn main() {
    println!("This is a code block");
    let x = 42;
    println!("x = {}", x);
}
```

Lists

Unordered List

- First item with **bold text**
- Second item with *italic text*
- Third item with a link

Ordered List

1. First ordered item
2. Second ordered item
3. Third ordered item

Blockquotes

> This is a simple blockquote with some text.

> This is a blockquote with a citation attribute.
>
> — https://example.com

> This is a blockquote with a citation attribute.
>
> — #asdf{}

Tables

Header 1	Header 2	Header 3
Cell 1	Cell 2	Cell 3
Cell 4	Cell 5	Cell 6

Definition Lists

Term 1 Definition for term 1

Term 2 Definition for term 2 with **bold text**

Subscript and Superscript

Here's some _{subscript text} and ^{superscript text}.

Line Breaks and Spacing

First line Second line

Third line Fourth line

Fifth line Sixth line

Horizontal Rules

HTML Entities

HTML entities: & < > " ‘ © ® ™

Nested Structures

This is a paragraph inside a div with **bold *and italic* text**.

- List item with link

Elements That Should Be Stripped

Video content Audio content Canvas content

Malformed HTML (Should Handle Gracefully)

<p>Unclosed paragraph Unclosed bold *Nested unclosed italic* <a href="https://example.com">Unclosed link

Special Characters in Attributes

Link with query params *Image: Image with "quotes" and & symbols*

Mixed Content

This paragraph has **bold**ed text, *italic*ized text, `code`-blocks, and links all mixed together.

> This is a blockquote with a paragraph inside it.
>
> - And a list item

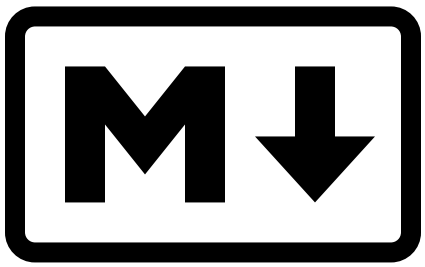
This should have quotes: “I am a quote”

Comments

Text after comment

Centered Text

This is centered



Complex Nested Structure

Section Title

Introduction paragraph with *emphasis*.

> Quote with **bold text** and a link.

Header	Another Header
Data	More data

Correctly escapes Typst syntax

@reference

@reference (`http://example.com`)

should be a plus + right there

#set

<label>

= this should not be a heading

this should not = be a heading

- item

/ Term: description

there should be backslash right here: \

~ there should be a tilde at the start of this line

#rect(width: 1cm)

#ad

/* block comment */

// line comment