Default title

Specification

This document specifies the syntax and structure of TypeDown, a lightweight markup language for creating structured text documents. TypeDown takes inspiration from Typst but focuses on being **embeddable** into your own projects and is better comparable to MarkDown.

Table of Contents

Basic Syntax

Characters TypeDown supports the standard Unicode character set.

Comments Comments start with an % and comment out everything until a newline.

Whitespace Whitespace characters (spaces, tabs, newlines) are generally ignored except within element content and for separating blocks.

Blocks A document in TypeDown consists of a sequence of blocks, each separated by two newline characters. Blocks can contain elements, text, or a combination of both.

Heading Block

Attributes Label

Content Model Block

Description

The Heading element defines a heading for a section of text. Heading levels are indicated by the number of consecutive equal signs ('=') at the beginning of the line, followed by a space. There are six heading levels (H1-H6), with a single '=' representing H1 and '=====' representing H6.

Example

```
= This is an H1 Heading

== This is an H2 Heading

This is some text content following the H2 Heading.

=== This is an H3 Heading

Another paragraph of text.
```

List

Attributes None

Content Model Block

Description

The List element defines a bulleted list. List items are created using the List-Item element. Where the first item **must** begin on a new line. List items can be further indented to create nested sub-lists.

Example

```
- This is the first list item
- This is the second list item
    - This is a nested list item under the second item
```

- This is the first list item
- This is the second list item
  - This is a nested list item under the second item

List-Item

Attributes Label

Content-Model Element

Description

List items represent a single item in a bullet list. Each item starts with an '-' followed by a space and its content.

Enum

Attributes None

Content Model Block

Description

The Enum element defines an ordered list. Enum items are created using the Enum-Item element. Where the first item **must** begin on a new line. Enum items can be further indented to create nested sub-lists. The visual numbering (1., 2., etc.) is for display purposes only and is not part of the markup itself.

Example

```
+ This is the first item in the ordered list
+ This is the second item
    + This is a nested item in the ordered list
```

1. This is the first item in the ordered list
2. This is the second item
    1. This is a nested item in the ordered list

Enum-Item

Attributes Label

Content-Model Element

Description

Enum items represent a single item in a ordered list. Each item starts with an '+' followed by a space and its content.

Term

Attributes None

Content-Model Block

Description

The term block consists out of several Term items

Example

```
> Cat: A small domestic mammal with soft fur. Cats are popular pets known for their
independence and hunting skills.
> Dog: A domesticated carnivorous mammal that is often kept as a companion animal.
Dogs have been bred in many varieties for various purposes such as hunting, herding,
and companionship.
```

Cat A small domestic mammal with soft fur. Cats are popular pets known for their independence and hunting skills.

Dog A domesticated carnivorous mammal that is often kept as a companion animal. Dogs have been bred in many varieties for various purposes such as hunting, herding, and companionship.

Term-Item

Attributes Label

Content-Model Element

Description

A term item starts with an '>' followed by the term name, a colon ':' and its description.

Table

Attributes None

Content-Model Block

Description

The Table element defines a table structure for organizing information in a grid of rows and columns. All table rows must have the same number of cells.

Example

```
| Header 1 | Header 2 | Header 3 |
| - This is a list item | + This is an ordered item | Text content |
| Row 2, Cell 1 | Row 2, Cell 2 | Row 2, Cell 3 |
```

Header 1	Header 2	Header 3
• This is a list item	1. This is an ordered item	Text content
Row 2, Cell 1	Row 2, Cell 2	Row 2, Cell 3

Table Row

Attributes Label

Content-Model Element

Description

A table row consists out of several cells, where each cell can either be:

- Plain text
- List-Item list_item
- Enum-Item enum_item

Raw

Attributes Label, Optional Language Identifier

Content-Model Block

Description

The Raw Block element defines a preformatted code block. The language identifier (if specified) can be used by the rendering tool to apply syntax highlighting. The content within the block is treated as verbatim text and is not interpreted by the markup language itself.

Example

```
fn main() {
  println!("Hello, world!");
}
```

Paragraph

Attributes None

Content-Model Block

Description

The Paragraph element represents a paragraph of text. Any text content that is not part of another block type is considered to be part of the default paragraph.

```
This is the first paragraph of text in the document.

Here is another paragraph of text, which can contain multiple lines.

Here is an example of a list following a paragraph:

- This is the first item in the list.
- This is the second item.

You can also mix and match other block types within your document.
```