

dtree

A Typst package for visualizing directory trees and file structures using indented text.

Published: Feb 20, 2026

dtree is a flexible and highly customizable directory tree visualization package for Typst. It renders directory structures using simple indented text, supports smart icon mapping, automatic styling rules, and vector-based connecting lines.

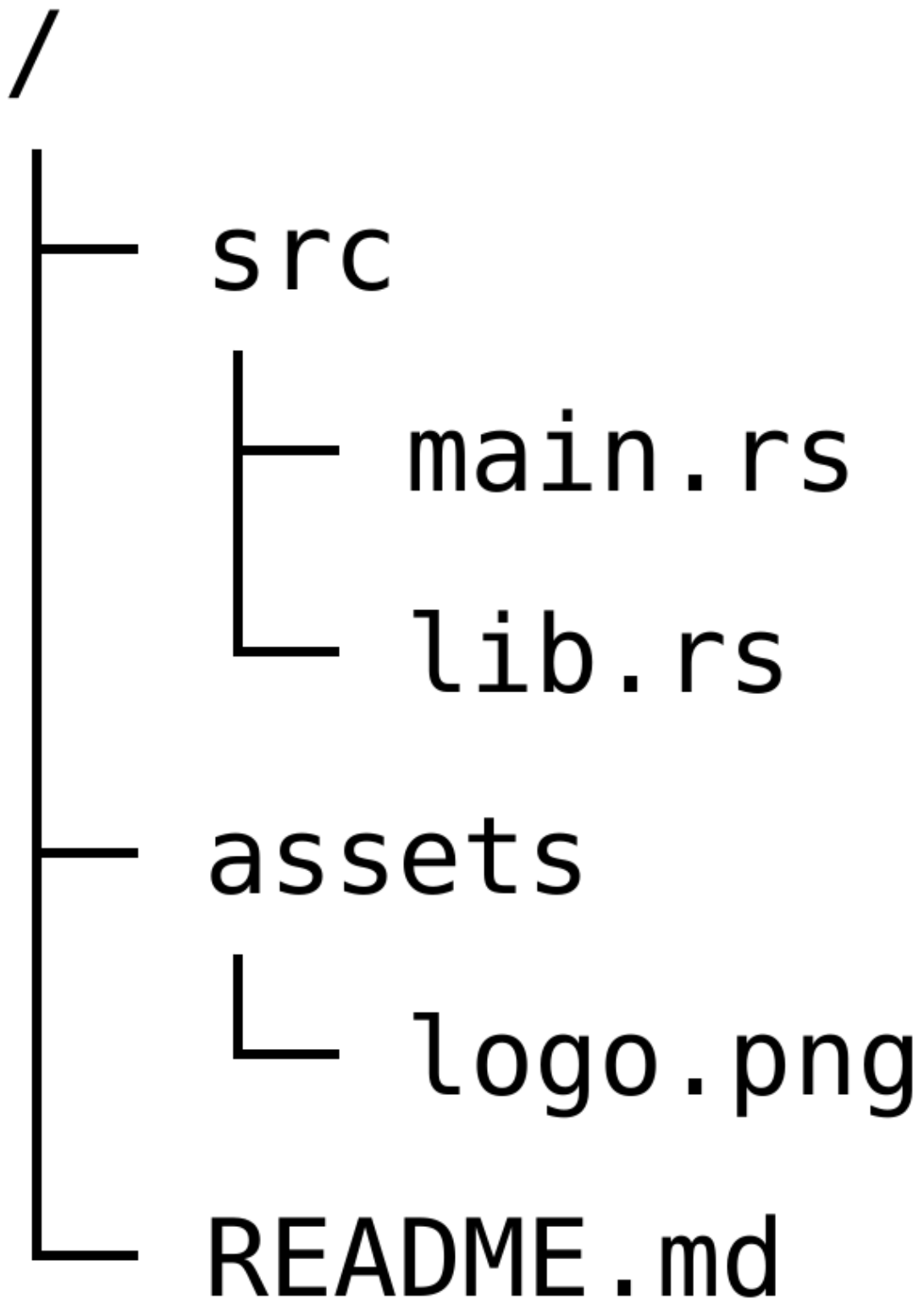
Usage

1. Basic Usage

The simplest way to use `dtree` is to pass a raw block. By default, it uses the standard monospace font defined in your document.

```
#import "@preview/dtree:0.1.0": dtree
```

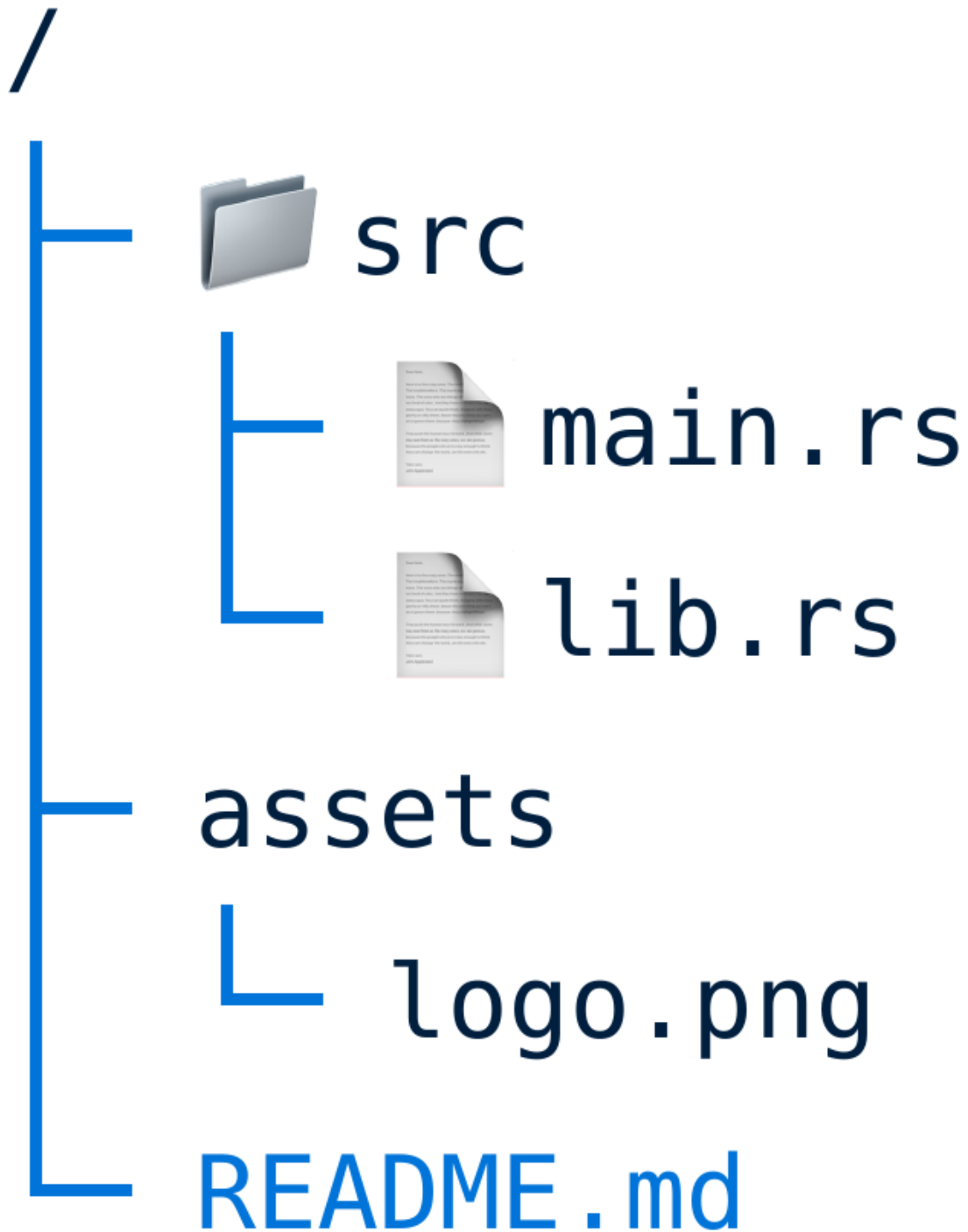
```
#dtree(``  
/  
  src  
    main.rs  
    lib.rs  
  assets  
    logo.png  
  README.md  
``)
```



2. Styling and Inline Parameters

You can customize the tree globally or per line. Use the `|` delimiter to specify an icon or parameters before the filename.

```
#dtree(  
  stroke: 1pt + blue,  
  fill: navy,  
  ``  
/  
  □ | src  
    □ | main.rs  
    □ | lib.rs  
assets  
  logo.png  
fill=blue | README.md  
  ``  
)
```



3. Advanced: Icon Rules & Images

To use images, you must read the file as bytes using `read(..., encoding: none)` and pass it to the icons dictionary.

Note: When using regex patterns, icon-rules must be an **array of pairs** (e.g., `((key, val),)`) instead of a dictionary.

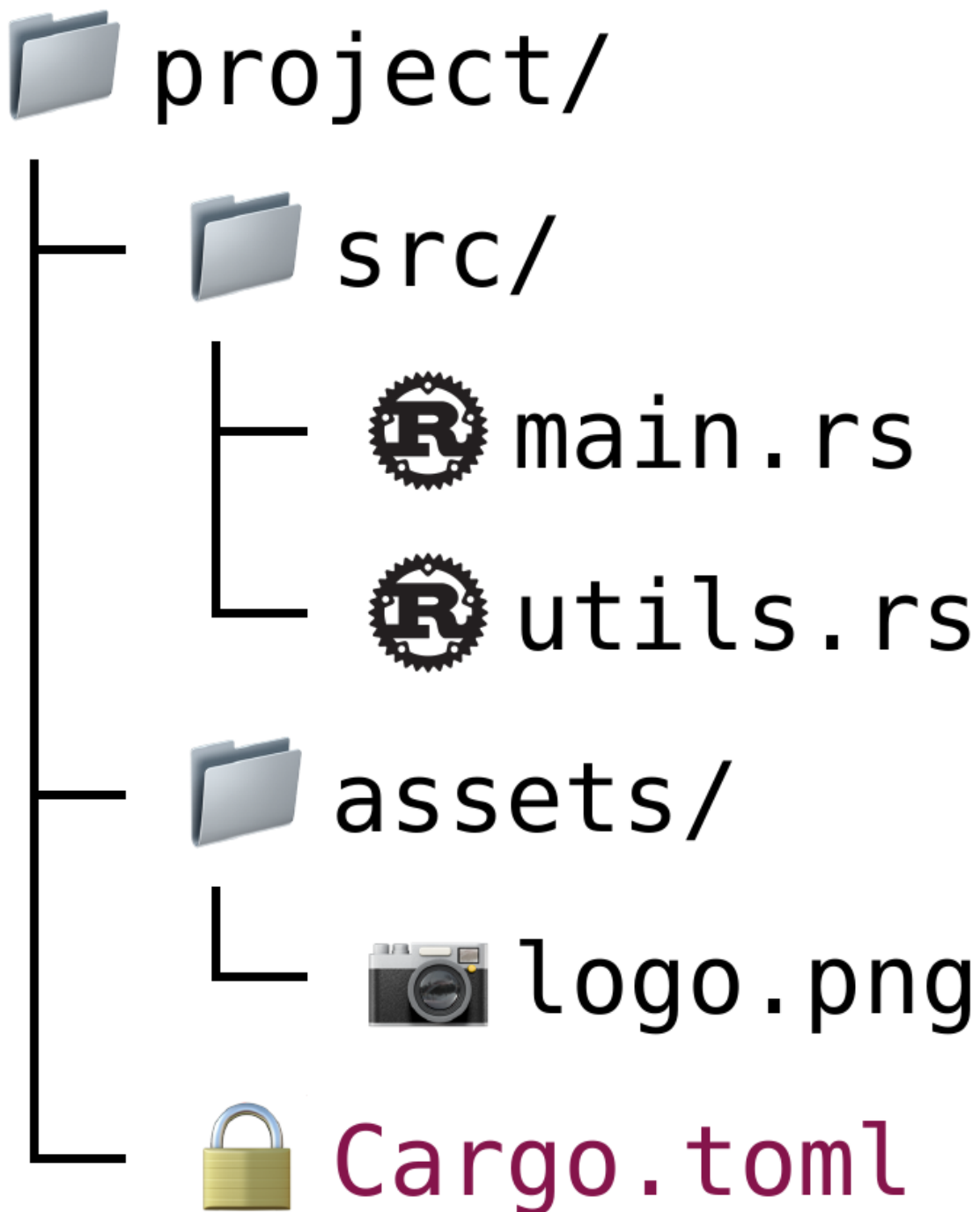
```
#let rust-icon = read("rust-logo.png", encoding: none)
#let config-icon = "🔧"
```

```

#let image-icon = "🖼️"
#let folder-icon = "📁"

#dtree(
  icons: (
    "rs": rust-icon,
    "img": image-icon,
    "dir": folder-icon,
  ),
  icon-rules: (
    (regex("/$"), "dir"),
    ("*.rs", (icon: "rs")),
    ("*.png", (icon: "img")),
    ("Cargo.toml", (icon: config-icon, fill: maroon)),
  ),
  ...
project/
src/
  main.rs
  utils.rs
assets/
  logo.png
Cargo.toml
...
)

```



Input Syntax

Each line in the raw block is parsed as follows:

[Icon Name], [Key=Value], ... | [Content]

1. **Icon Name** (Optional):

- A key defined in the icons dictionary.
- A raw emoji or text string (e.g., 📁, Rs).
- **Note:** Direct file paths (e.g. image.png) are not supported inside the raw block. Please load images via the icons dictionary.

2. **Parameters** (Optional): Comma-separated key=value pairs.
 - Supported keys: size, dx, dy, fill (or color), font.
3. **Delimiter**: The | character separates metadata from the content name.

Examples:

- `📄 | Documents` (Icon only)
- `fill=red | Important.txt` (Params only)
- `my_icon, size=1.5em, dy=2pt | image.png` (Icon + Params)
- `file.txt` (Content only)

Icon Priority

Icons are resolved in the following order (highest to lowest priority):

1. **Inline specification** — Direct icon specified in the line (e.g., `📄 | sushi.txt`)
2. **icon-rules** — Pattern-based rules that match the content name
3. **default-icon** — Fallback icon if no other match is found

API Reference

dtree

Argument	Type	Default	Description
body	content or str	Required	The directory structure text. Using a raw block is recommended to preserve indentation.
Layout	indent-width	length	1.5em
The horizontal width of one indentation level.	row-height	length	1.5em
The height of a single row.	spacing	length	0.5em
The space between the tree lines and the content.	indent-marker	str	" "
The character used to calculate indentation levels (e.g., set to "\t" for tabs or " " for 2-space indentation).	Style	stroke	stroke
0.7pt + black	The style of the connecting lines (thickness + color).	font	str or array
none	The font family. If none , uses raw() (inherits code font). If specified, uses text(font: ...) .	size	length
10pt	The font size for filenames.	fill	color
black	The text color for filenames.	Icons	icons
dictionary	(:)	A dictionary mapping names to content or bytes (loaded via read(..., encoding: none)).	icon-rules

Argument	Type	Default	Description
array	()	An array of pairs mapping patterns (Glob *.ext or regex) to an icon name or a style dictionary. Example: ((regex("."), val),) .	default-icon
content	none	An icon to display if no specific icon or rule matches.	icon-size
length	1em	The default size for icons.	icon-dx
length	-1pt	Horizontal offset for icons.	icon-dy
length	0pt	Vertical offset for icons.	

Parameter Dictionary (icon-rules & Inline Params)

When defining icon-rules values or using inline syntax, the following keys are available:

- icon: (String) The name of the icon to use (only for icon-rules).
- size: (Length) Size of the icon/image.
- fill or color: (Color) Text color.
- font: (String) Font family.
- dx: (Length) Horizontal offset.
- dy: (Length) Vertical offset.

License

This project is distributed under the MIT License. See LICENSE for details.