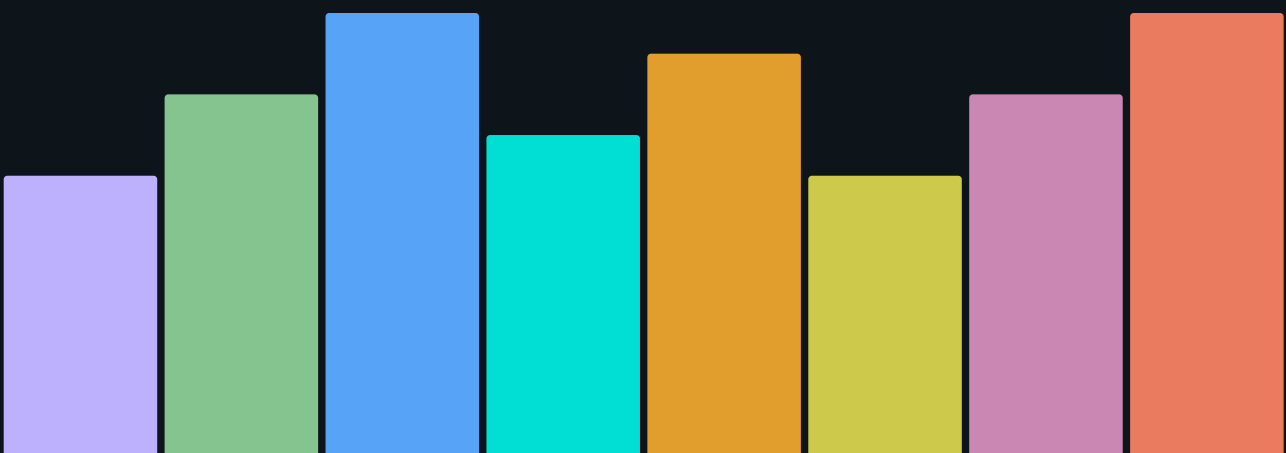
KVILE

Christoph Pilka · 2026

Type for code

Choosing and setting a monospace font, with measurements



Type for code

Choosing and setting a monospace font, with measurements

Copyright 2026 Christoph Pilka.
A companion to *Color that lets your eyes rest.* All font numbers in this book are measured from the font files (Fontsource 5.x, Latin subsets, regular weight) with a script in the Kvile repository.

The fonts shown are licensed under the SIL Open Font License 1.1 and are embedded here with their licenses listed in Appendix B. This book is not medical advice. Where the research is thin, the text says so.

Contents

01 What a code font has to do

02 What the research says, and what it does not

03 Measuring eight fonts

04 Eight fonts, one sample

05 One font, four variants

06 Size, line height and spacing

07 Weight, bold and italic

08 Ligatures and texture healing

09 Licenses and where to get the fonts

10 Setting it up

APPENDICES

A All measurements

B Font notices

C Glossary and sources

Before you start

A color theme only works as well as the letters it is painted on. Thin strokes, a small x-height or cramped line spacing can undo careful contrast, and a font with confusing characters costs time no matter how good the palette is.

This book is short and practical. It explains what a coding font has to do, what research does and does not say about it, how eight popular open fonts compare when you measure them, and how to set size, line height, weights, italics and ligatures in the tools Kvile supports. It ends with a recommendation and a checklist.

A note on honesty: **no study I found ranks coding fonts head to head.** The research that exists is about neighboring questions (identifier style, display polarity, text spacing), and the book says exactly how far each result reaches. Everything measurable is measured. Everything else is labeled as judgment.

> IN PRACTICE
>
> Treat the numbers as a way to narrow the choice, then try your top two for a week. Your eyes, your screen and your room are part of the test.

01

What a code font has to do

Code is not read like prose. It is scanned for shapes, and every character that looks like another one costs a moment.

Reading by shape

You rarely read code word by word. You jump between identifiers, brackets and operators, and your eyes land on shapes first. That makes *distinguishability* the first job of a code font: the digit zero against the letter O, the digit one against a lowercase `l` and a capital I, `rn` against `m`, a backtick against a quote, curly braces against parentheses. Figure 3 sets these pairs in all eight fonts of this book.

Fixed pitch

In a monospace font every glyph has the same advance width. That is what keeps columns aligned in a terminal and makes indentation visible. In six of the eight fonts measured here the advance is 0.6 em (Cascadia Code is 0.586 em and Monaspace Neon is 0.620 em). At 14 pt that is 8.4 pt per column, so a 120-column line needs 1008 pt of width; at 13 pt it needs 936 pt.

The price of fixed pitch is rhythm. An `i` must fill the same cell as an `m`, so type designers add serifs and slabs to narrow letters, and a monospace font can look gappy next to a proportional one. Some newer fonts try to compensate. Chapter 8 returns to this.

Five jobs

1. **Tell similar characters apart.** Chapter 4.
2. **Keep a steady rhythm** in a grid of equal cells.
3. **Survive small sizes** on real screens: a tall x-height and open shapes. Chapter 3 measures this.
4. **Offer weights and a real italic** for emphasis. Kvile's VS Code theme sets comments in italic and Markdown headings and bold text in bold, so the font must have both. Chapter 7.
5. **Work with color.** Syntax colors sit on letter strokes; thin or small strokes leave less room for color to show. Chapters 2 and 6.

02

What the research says, and what it does not

There is useful evidence on neighboring questions. There is no study that crowns a coding font.

No head-to-head ranking

In the sources checked for this book I found no controlled study that compares coding fonts against each other for speed, accuracy or fatigue. Font articles and listicles are plentiful, but they are opinion. What follows is what *can* be said, with its limits.

Identifier style: small effects, trade-offs

Binkley, Davis, Lawrie, Maletic, Morrell and Sharif (2012) compared camelCase with under_score identifiers in five studies with about 150 participants. Camel case often gave higher accuracy, and in one study raised the odds of a correct answer by about 51%. It was often slower: underscore paragraphs in one reading study were read about 2.7 seconds faster (a marginally significant result), and eye tracking showed more effort on long camel-case identifiers. Experienced developers appeared less affected than beginners.

This is not about fonts. It matters here because it shows what readability effects in code look like: small, mixed, and a trade between speed and accuracy. Treat anyone who promises a large effect from a font with caution.

Display polarity: brightness is the driver

Buchner, Mayr and Brandt (2009) reported that the advantage of dark text on a light background (positive polarity) is due to high display luminance. Piepenbrock, Mayr and Buchner (2014) found smaller pupil size and better proofreading performance with positive than with negative polarity displays. Mathôt's 2014 summary of this work argues that the effect is really a small-pupil advantage, that performance improves with display brightness whatever the polarity, and that a well-lit room and regular breaks help. He also says that some truth to bright-screen fatigue is plausible but less established.

For a dark theme such as Kvile Natt, the careful reading is this: the studies tested reading tasks, not code fonts, and they do not say dark themes are wrong. They suggest that with light text on a dark ground, small and thin strokes have the least margin. That is an *inference*, not a result, and it is the reason this book favors regular weights and generous x-heights for dark variants.

Text spacing: a floor, not a recommendation

WCAG 2.2 success criterion 1.4.12 requires that web content keeps working when a user overrides spacing to at least 1.5 times the font size for line height, 2 times for paragraph spacing, 0.12 times for letter spacing and 0.16 times for word spacing. It does not require authors to use these values as defaults. The numbers are a tolerance test: they say what content must survive, not what is most comfortable.

> IN PRACTICE
>
> Use research to decide what to distrust. Use measurements to narrow choices. Use a week of real work to decide.

03

Measuring eight fonts

Size, width and line height differ more than the point size on the label suggests.

Method

I downloaded the Latin subsets of eight open fonts from the Fontsource packages (version 5.x) and read their tables with the fontTools library: units per em, the advance of the digit 0, the height of the lowercase x and the capital H (measured from the glyph outlines), and the natural line height, which is the sum of the `hhea` ascent, descent and line gap. Everything is divided by the em, so numbers compare across fonts. The script is `ebook-type/metrics.py` in the repository.

Natural line height is what many editors and terminals use when no line height is set. Some apps read other metric tables, so treat it as a guide to how roomy a font is by default, not a promise.

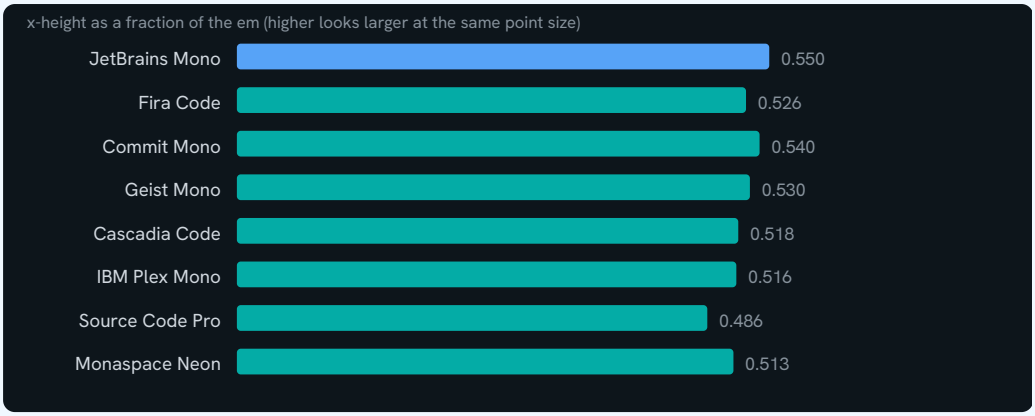


Figure 1. Lowercase height in em. JetBrains Mono is tallest at 0.550; Source Code Pro is lowest at 0.486, a difference of 13%.

What the numbers say

- **Same point size, different looks.** At 14 pt, JetBrains Mono has lowercase letters 13% taller than Source Code Pro. A font that looks small is often just low in x-height.
- **Line height is not neutral.** The natural line height runs from 1.10 (Commit Mono) to 1.32 (JetBrains Mono), so "normal" line spacing differs by about 20% between fonts.
- **Width.** Six fonts use 0.6 em. Cascadia Code is slightly narrower (0.586), which fits about 2.4% more columns in the same width, and Monaspace Neon is wider (0.620).
- **x-height to cap-height.** The ratios sit between 0.70 and 0.77, so lowercase-heavy code looks quite different from ALL_CAPS constants depending on the font.



Figure 2. Defaults differ by font. Set the line height yourself so changing fonts does not silently change your layout.

04

Eight fonts, one sample

The same code, in the same Kvile Natt colors, in each font.

All eight fonts below are free under the SIL Open Font License and are set at the same size. Look at the shape of the lowercase letters, the width of the strokes, how the digits and punctuation read, and how the comment in the fifth line looks.

JetBrains Mono

```
import { createServer } from "node:http";

type Session = { hours: number };

// Remind people to take a break.
const MAX_HOURS = 8;

export function nextBreak(s: Session) {
  if (s.hours > MAX_HOURS) return 0;
  return Math.min(90, 96 - s.hours * 12);
}
```

Fira Code

```
import { createServer } from "node:http";

type Session = { hours: number };

// Remind people to take a break.
const MAX_HOURS = 8;

export function nextBreak(s: Session) {
  if (s.hours > MAX_HOURS) return 0;
  return Math.min(90, 96 - s.hours * 12);
}
```

Commit Mono

```
import { createServer } from "node:http";

type Session = { hours: number };

// Remind people to take a break.
const MAX_HOURS = 8;

export function nextBreak(s: Session) {
  if (s.hours > MAX_HOURS) return 0;
  return Math.min(90, 96 - s.hours * 12);
}
```

Geist Mono

```
import { createServer } from "node:http";

type Session = { hours: number };

// Remind people to take a break.
const MAX_HOURS = 8;

export function nextBreak(s: Session) {
  if (s.hours > MAX_HOURS) return 0;
  return Math.min(90, 96 - s.hours * 12);
}
```

Cascadia Code

```ts
import { createServer } from "node:http";

type Session = { hours: number };

// Remind people to take a break.
const MAX_HOURS = 8;

export function nextBreak(s: Session) {
  if (s.hours > MAX_HOURS) return 0;
  return Math.min(90, 96 - s.hours * 12);
}
```

IBM Plex Mono

```ts
import { createServer } from "node:http";

type Session = { hours: number };

// Remind people to take a break.
const MAX_HOURS = 8;

export function nextBreak(s: Session) {
  if (s.hours > MAX_HOURS) return 0;
  return Math.min(90, 96 - s.hours * 12);
}
```

Source Code Pro

```ts
import { createServer } from "node:http";

type Session = { hours: number };

// Remind people to take a break.
const MAX_HOURS = 8;

export function nextBreak(s: Session) {
  if (s.hours > MAX_HOURS) return 0;
  return Math.min(90, 96 - s.hours * 12);
}
```

Monaspace Neon

```ts
import { createServer } from "node:http";

type Session = { hours: number };

// Remind people to take a break.
const MAX_HOURS = 8;

export function nextBreak(s: Session) {
  if (s.hours > MAX_HOURS) return 0;
  return Math.min(90, 96 - s.hours * 12);
}
```

JetBrains Mono	0 O o 1 l I | rn m { } ( ) [ ] ` ' "
Fira Code	0 O o 1 l I | rn m { } ( ) [ ] ` ' "
Commit Mono	0 O o 1 l I | rn m { } ( ) [ ] ` ' "
Geist Mono	0 O o 1 l I | rn m { } ( ) [ ] ` ' "
Cascadia Code	0 O o 1 l I | rn m { } ( ) [ ] ` ' "
IBM Plex Mono	0 O o 1 l I | rn m { } ( ) [ ] ` ' "
Source Code Pro	0 O o 1 l I | rn m { } ( ) [ ] ` ' "
Monaspace Neon	0 O o 1 l I | rn m { } ( ) [ ] ` ' "

Figure 3. Look-alike characters in each font: zero and O, one and l and I, rn and m, brackets, quotes. Judge them at your real size, not here.

Notes on each font

FONT	FROM	WEIGHTS IN THIS BOOK'S SOURCE	ITALIC	LIGATURES (CALT FEATURE)
JetBrains Mono	JetBrains	100-800	yes	yes
Fira Code	Nikita Prokopov, from Mozilla and Telefonica's Fira Mono	300-700	no	yes
Commit Mono	Eigil Nikolajsen	200-700	yes	yes
Geist Mono	Vercel	100-900	yes	none found
Cascadia Code	Microsoft	200-700	yes	yes
IBM Plex Mono	IBM	100-700	yes	none found
Source Code Pro	Adobe	200-900	yes	none found
Monaspace Neon	GitHub Next	200-800	yes	yes

- **JetBrains Mono.** Eight styles from Thin to ExtraBold, each with an italic. It is built to maximize lowercase height while keeping standard character width, and a no-ligature edition (NL) exists.
- **Fira Code.** Extends Fira Mono with ligatures for operators such as →, ⩽ and :=. The project stresses that this is a rendering effect and the underlying code stays ASCII. In the Fontsource

distribution used here it has no italic files, so an italic comment is a synthesized slant.

- **Commit Mono.** A variable family from weight 200 to 700, upright and italic, deliberately understated so attention stays on the code.
- **Geist Mono.** From Vercel, part of the Geist family; licensed under OFL 1.1. No ligature feature was found in the files measured.
- **Cascadia Code.** Microsoft's font for Windows Terminal and editors since 2019. Cascadia Code has ligatures; Cascadia Mono does not; versions with Powerline (PL) and Nerd Font (NF) symbols exist.
- **IBM Plex Mono.** The monospace member of IBM's Plex superfamily, with weights from 100 to 700 and italics.
- **Source Code Pro.** Adobe's open monospace font, with the lowest x-height of the eight (0.486 em) and a wide weight range.
- **Monaspace Neon.** One of five families (Neon, Argon, Xenon, Radon, Krypton) that share metrics so they can be mixed. It introduces "texture healing", which adjusts spacing around characters through the `calt` feature, and ten stylistic sets for ligature groups.

Not shown. Berkeley Mono is a commercial font from U.S. Graphics Company (licenses from $75; its terms limit use in IDEs and terminal apps and ask you to contact the company, so read them first). SF Mono comes with Apple's developer tools, but Apple restricts redistribution of its SF fonts, so do not embed it in a web page or an app.

05

One font, four variants

The same font looks different on a dark ground and on a light one.

Kvile ships four variants: Natt (night), Skumring (dusk), Dag (day) and Kontrast (high contrast). The recommended font is the same in all four. Here is the sample in JetBrains Mono on each ground.

JetBrains Mono

```
import { createServer } from "node:http";

type Session = { hours: number };

// Remind people to take a break.
const MAX_HOURS = 8;

export function nextBreak(s: Session) {
  if (s.hours > MAX_HOURS) return 0;
  return Math.min(90, 96 - s.hours * 12);
}
```

JetBrains Mono · Kvile Skumring

```
import { createServer } from "node:http";

type Session = { hours: number };

// Remind people to take a break.
const MAX_HOURS = 8;

export function nextBreak(s: Session) {
  if (s.hours > MAX_HOURS) return 0;
  return Math.min(90, 96 - s.hours * 12);
}
```

JetBrains Mono · Kvile Dag

```
import { createServer } from "node:http";

type Session = { hours: number };

// Remind people to take a break.
const MAX_HOURS = 8;

export function nextBreak(s: Session) {
  if (s.hours > MAX_HOURS) return 0;
  return Math.min(90, 96 - s.hours * 12);
}
```

JetBrains Mono · Kvile Kontrast

```
import { createServer } from "node:http";

type Session = { hours: number };

// Remind people to take a break.
const MAX_HOURS = 8;

export function nextBreak(s: Session) {
  if (s.hours > MAX_HOURS) return 0;
  return Math.min(90, 96 - s.hours * 12);
}
```

What to look for

- **Stroke weight.** Some readers see a glow around bright text on a dark ground (halation, see the color book), so a dark variant can feel heavier than a light one at the same weight. As a rule of thumb, not a measured result: if a dark variant feels heavy, try a lighter weight; if it feels thin, move up one step.
- **Small text.** Comments and line numbers are drawn in quieter colors, so they depend most on x-height.
- **Kontrast.** The highest contrast of the four. Check that regular does not feel too heavy there, and that bold and regular still look different.

> IN PRACTICE
>
> Test your font in the variant you use most, at your real size, in your real room light. A specimen sheet in a different variant tells you less than ten minutes in your own editor.

06

Size, line height and spacing

Point size is only half of how big text looks.

Match the x-height, not the number

Because fonts differ in x-height, the same point size gives different lowercase sizes. If you like JetBrains Mono at 14 pt (x-height 0.550 em, or 7.7 pt), this is the size that gives the same lowercase height in the other fonts:

FONT	X-HEIGHT (EM)	SIZE MATCHING JETBRAINS MONO 14 PT
JetBrains Mono	0.550	14.0 pt
Fira Code	0.526	14.6 pt
Commit Mono	0.540	14.3 pt
Geist Mono	0.530	14.5 pt
Cascadia Code	0.518	14.9 pt
IBM Plex Mono	0.516	14.9 pt
Source Code Pro	0.486	15.8 pt
Monaspace Neon	0.513	15.0 pt

Matching x-height keeps the look of the text the same, but not the width: a font with a lower x-height set larger also uses more columns, because the advance grows with the size.

Line height

Line height is the space between baselines, as a multiple of the font size. Too tight and descenders touch ascenders, too loose and code loses its block shape. The figure sets the same code at three values.

line height 1.1

```
function nextBreak(hours) {
  if (hours > 8) return 0;
  return Math.min(90, 96 - hours * 12);
}
```

line height 1.3

```
function nextBreak(hours) {
  if (hours > 8) return 0;
  return Math.min(90, 96 - hours * 12);
}
```

line height 1.5

```
function nextBreak(hours) {
  if (hours > 8) return 0;
  return Math.min(90, 96 - hours * 12);
}
```

Figure 4. The same code at line heights 1.1, 1.3 and 1.5.

Kvile's practical starting points are about 1.1 to 1.2 in terminals (so box-drawing characters and Powerline symbols still connect) and 1.4 to 1.5 in editors. These are working defaults, not research results. WCAG's 1.5 is a tolerance test for web content, so it is a convenient reference point for editors, not a target.

Kvile's size

> KVILE DECISION
>
> Kvile recommends **JetBrains Mono at 14 pt**. It has the tallest x-height of the eight fonts measured, a roomy natural line height of 1.32, and it is free under the OFL. At 14 pt, 120 columns need about 1008 points of width.

Two more rules: set the size in your *editor and terminal* and keep the macOS display scaling the same when you compare fonts; and prefer zooming a window (Cmd and plus) for presentations instead of changing the saved size.

07

Weight, bold and italic

A theme that uses italics and bold needs a font that has them.

Regular is the workhorse

Use weight 400 for code. Thin and light weights lose strokes at small sizes and, on dark grounds, give color the least to work with. Reserve bold for the few places where a theme asks for it.

What Kvile styles

In the VS Code theme, comments are italic, and Markdown headings and bold text are bold; Markdown italic and quotes are italic. Everything else is regular. So a font used with Kvile needs a working italic for comments, and a bold close in color weight to the regular.

Does the font have a real italic?

Of the eight fonts, seven have italic files in the Fontsource distribution used here. Fira Code does not, so editors show a synthesized slant. A synthesized italic is only an upright font tilted, which can look uneven.

> IN PRACTICE
>
> If you dislike italics in code, turn them off in the theme or in your editor (for VS Code, a `editor.tokenColorCustomizations` rule for comments), not by changing fonts.

JetBrains Mono	*// Remind people to take a break.*
Fira Code no italic file	*// Remind people to take a break.*
Commit Mono	*// Remind people to take a break.*
Geist Mono	*// Remind people to take a break.*
Cascadia Code	*// Remind people to take a break.*
IBM Plex Mono	*// Remind people to take a break.*
Source Code Pro	*// Remind people to take a break.*
Monaspace Neon	*// Remind people to take a break.*

Figure 5. The Kvile comment style in each font (italic, in the Natt comment color). Fira Code has no italic file here, so its comment is slanted by the browser.

08

Ligatures and texture healing

Some fonts draw two or three characters as one shape. Whether you want that is taste.

What they are

A *ligature* replaces a sequence such as → or ≠ with a single designed glyph. The file still contains the original characters; only the drawing changes. Fira Code's project says exactly this: it is a rendering feature and the code stays ASCII-compatible. Cascadia Code has the feature in its Code edition and leaves it out of Cascadia Mono; JetBrains Mono offers a no-ligature edition called NL.

Texture healing, introduced by Monaspace, is different. It lets characters borrow or give width to neighbors through the `calt` feature to even out the rhythm of a monospace font, so a narrow `i` and a wide `m` look less gappy.

For and against

- **For:** operators read as one symbol; less visual noise; some people simply like it.
- **Against:** an unusual glyph can hide what you typed (for example in a diff or a screenshot someone else will read); copying from a screenshot is harder; some tools render them differently.

Switching them on or off

TOOL	SETTING
VS Code	`"editor.fontLigatures": true`
kitty	`disable_ligatures never | cursor | always`
WezTerm	`config.harfbuzz_features = { 'calt=0', 'clig=0', 'liga=0' }`

The kitty option accepts `never` (the default, ligatures always drawn), `cursor` (not drawn under the cursor, which makes editing easier) and `always` (never drawn). For other tools, check the documentation for "ligatures" or "font features"; the setting names change between versions.

> KVILE DECISION
>
> Kvile's colors do not depend on ligatures. Its screenshots are taken in JetBrains Mono, which has them as an option. Choose by taste.

09

Licenses and where to get the fonts

Free to use is not always free to bundle.

FONT	LICENSE	COST	MAY YOU BUNDLE IT?
JetBrains Mono	SIL OFL 1.1	free	yes, keep the license and copyright notice
Fira Code	SIL OFL 1.1	free	yes, keep the license and copyright notice
Commit Mono	SIL OFL 1.1	free	yes, keep the license and copyright notice
Geist Mono	SIL OFL 1.1	free	yes, keep the license and copyright notice
Cascadia Code	SIL OFL 1.1	free	yes, keep the license and copyright notice
IBM Plex Mono	SIL OFL 1.1	free	yes, keep the license and copyright notice
Source Code Pro	SIL OFL 1.1	free	yes, keep the license and copyright notice
Monaspace Neon	SIL OFL 1.1	free	yes, keep the license and copyright notice
Berkeley Mono	commercial	from $75	no, read the terms; use in IDEs and terminal apps needs to be cleared with the publisher
SF Mono	Apple	comes with Apple's developer tools	no, Apple restricts redistribution

What the OFL allows

The SIL Open Font License lets you use, study, copy, embed and redistribute a font, including in commercial work, as long as the copyright notice and license stay with it and you do not sell the font on its own. Some fonts reserve a name, so a modified version must be renamed. This book embeds Latin subsets of the eight fonts under these terms; Appendix B lists the notices.

Where to get them

- **macOS (Homebrew):** for example `brew install --cask font-jetbrains-mono`.
- **JetBrains IDEs** ship JetBrains Mono since version 2019.3.
- **Websites:** the Fontsource packages on npm give self-hostable files for every font in this book.
- **Windows:** Cascadia Code is bundled with Windows Terminal and Windows 11.

10

Setting it up

A short recommendation, the settings for the tools Kvile supports, and a checklist.

The recommendation

> KVILE DECISION
>
> **JetBrains Mono, regular weight, 14 pt**, line height about 1.5 in editors and 1.15 to 1.2 in terminals. If you prefer another font, pick one from this book, set it so its x-height matches (Chapter 6), make sure it has a real italic (Chapter 7), and choose ligatures by taste (Chapter 8).

Settings

kitty

```
font_family JetBrains Mono
font_size 14.0
modify_font cell_height 115%
```

Ghostty

```
font-family = JetBrains Mono
font-size = 14
```

Alacritty

```
[font]
size = 14
[font.normal]
family = "JetBrains Mono"
[font.offset]
y = 3
```

WezTerm

```
config.font = wezterm.font("JetBrains Mono")
config.font_size = 14
```

VS Code

```
"editor.fontFamily": "'JetBrains Mono', monospace",
"editor.fontSize": 14,
"editor.lineHeight": 1.5
```

Zed

```
"buffer_font_family": "JetBrains Mono",
"buffer_font_size": 14
```

Neovide

```
vim.o.guifont = "JetBrains Mono:h14"
```

Checklist

1. Pick two fonts from Chapter 4 and set them to the same x-height (Chapter 6).
2. Use regular weight (400) for code and one step heavier for bold.
3. Set line height yourself in every tool, so a font change does not move your layout.
4. Check that your font has a real italic if your theme uses italics.
5. Decide on ligatures once and set it in every tool.
6. Use each font for a week in the variant you use most.
7. Keep a well-lit room and take breaks. The research that exists points at brightness and breaks more than at any font.

A

Appendix A: all measurements

Regular weight, Latin subset, from the Fontsource 5.x packages. Everything is in em.

FONT	UNITS/EM	ADVANCE OF 0	X-HEIGHT	CAP HEIGHT	X/CAP	ASCENT	DESCENT	LINE GAP	NATURAL LINE HEIGHT
JetBrains Mono	1000	0.600	0.550	0.730	0.753	1.020	0.300	0.000	1.320
Fira Code	2000	0.600	0.526	0.689	0.765	0.990	0.322	0.000	1.312
Commit Mono	1000	0.600	0.540	0.700	0.771	0.900	0.200	0.000	1.100
Geist Mono	1000	0.600	0.530	0.710	0.746	1.005	0.295	0.000	1.300
Cascadia Code	2048	0.586	0.518	0.693	0.746	0.928	0.234	0.000	1.162
IBM Plex Mono	1000	0.600	0.516	0.698	0.739	1.025	0.275	0.000	1.300
Source Code Pro	1000	0.600	0.486	0.656	0.741	0.984	0.273	0.000	1.257
Monaspace Neon	2000	0.620	0.513	0.730	0.703	0.782	0.217	0.200	1.200

Ascent, descent and line gap are the `hhea` values. Apps that use the OS/2 typographic or Windows metrics instead may show different spacing. Ligature flags: the `calt` feature was found in JetBrains Mono, Fira Code, Commit Mono, Cascadia Code, Monaspace Neon.

B

Appendix B: font notices

All eight fonts are licensed under the SIL Open Font License, Version 1.1 (openfontlicense.org). Copyright lines from each package's license file.

- **JetBrains Mono**. Copyright 2020 The JetBrains Mono Project Authors (https://github.com/JetBrains/JetBrainsMono)
- **Fira Code**. Copyright 2014-2020 The Fira Code Project Authors (https://github.com/tonsky/FiraCode)
- **Commit Mono**. Copyright 2023 Eigil Nikolajsen (github.com/eigilnikolajsen/commit-mono). Licensed under the SIL Open Font License 1.1.
- **Geist Mono**. Copyright 2024 The Geist Project Authors (https://github.com/vercel/geist-font.git)
- **Cascadia Code**. Cascadia Code, by Microsoft (github.com/microsoft/cascadia-code). The packaged license file names no copyright holder line; see the project repository. Licensed under the SIL Open Font License 1.1.
- **IBM Plex Mono**. Copyright 2017 IBM Corp. All rights reserved.
- **Source Code Pro**. Source Code Pro, by Adobe (github.com/adobe-fonts/source-code-pro). The packaged license file names no copyright holder line; see the project repository. Licensed under the SIL Open Font License 1.1.
- **Monaspace Neon**. Copyright (c) 2023, GitHub https://github.com/githubnext/monaspace with Reserved Font Name "Monaspace", including subfamilies: "Argon", "Neon", "Xenon", "Radon", and "Krypton"

C

Appendix C: glossary and sources

Glossary

Advance width	The horizontal space a glyph takes. Equal for all glyphs in a monospace font.
calt	OpenType feature (contextual alternates) used for ligatures and texture healing.
Cap height	Height of a capital letter such as H.
Em	The size of the font; 1 em equals the font size.
Ligature	One designed glyph shown for a sequence of characters.
OFL	SIL Open Font License, a free license for fonts.
Polarity	Dark text on light (positive) or light text on dark (negative).
Texture healing	Spacing adjustments between neighbors in a monospace font to even out rhythm.
x-height	Height of a lowercase letter such as x, without ascenders.

Sources

- Binkley, D., Davis, M., Lawrie, D., Maletic, J. I., Morrell, C., Sharif, B. (2012). The impact of identifier style on effort and comprehension. *Empirical Software Engineering*.
- Buchner, A., Mayr, S., Brandt, M. (2009). The advantage of positive text-background polarity is due to high display luminance. *Ergonomics*, 52(7), 882-886.
- Piepenbrock, C., Mayr, S., Buchner, A. (2014). Smaller pupil size and better proofreading performance with positive than with negative polarity displays. *Ergonomics*.
- Mathôt, S. (2014). Is bright text on a dark background a good idea? cogsci.nl/blog/is-bright-text-on-a-dark-background-a-good-idea
- W3C. Understanding Success Criterion 1.4.12: Text Spacing, WCAG 2.2. w3.org/WAI/WCAG22/Understanding/text-spacing
- JetBrains Mono, github.com/JetBrains/JetBrainsMono. Monaspace, github.com/githubnext/monaspace. Geist, github.com/vercel/geist-font. Cascadia Code, en.wikipedia.org/wiki/Cascadia_Code. Commit Mono and Fira Code pages on fontsource.org. Berkeley Mono, usgraphics.com/products/berkeley-mono.
- WezTerm font shaping, wezterm.org/config/font-shaping.html. kitty.conf manual, sw.kovidgoyal.net/kitty/conf/.
- Measurements: `ebook-type/metrics.py` and `ebook-type/metrics.json` in the Kvile repository.

- The Kvile color book, *Color that lets your eyes rest*, covers contrast, perception and syntax color in depth.