PAPER · NOVEMBER 2026

Separate Families, Explicit Choices

Slides ↗ Transcript ↗ Paper ↗ Observation data ↗ Interactive demo ↗

Download paper PDF

Summary
Disclosure
Prefer the font or CSS when sufficient
Separate names need an authored relationship
What the default example selects
Hysteresis makes selection depend on the route
Historical context without a market story
Objections answered
Recommendations
Open questions
Method and verification boundary
Sources

Summary

When optical cuts ship as separate CSS families, web authors should publish an explicit selection policy rather than expect browser optical sizing to discover the relationship. opszStepper demonstrates one way to execute a size-to-family policy when observed events trigger a computed-size recheck. It supplies the routing, not the typographic judgment.

Start with the alternatives. For a suitable variable font with an opsz axis, use browser optical sizing when its automatic policy fits. For separate families tied to stable editorial roles, ordinary CSS assignments are often enough. Use a computed-size switch when following computed size is the actual requirement. This paper argues for an inspectable handoff between foundry and author, not universal adoption of a library.

Disclosure

Overpunch makes opszStepper. Quinn Keaveney speaks from Vancouver and has spent 15+ years building websites for type foundries. Overpunch is also building Typetin, a storefront for independent foundries, currently in development. Those interests inform this proposed authoring practice; they are not independent evidence of demand or outcomes.

Research began on 6 November 2026; the sources and argument were reviewed on 8 November. The script, slides and paper serve different purposes: the paper includes qualifications and methods that a short spoken presentation cannot carry in full.

Prefer the font or CSS when sufficient

For a font that provides an optical size axis, the high-level CSS route is font-optical-sizing. The initial auto policy permits optical adaptation; the font and platform participate in selection. Explicit low-level variation settings can take precedence, so check existing overrides before concluding the feature is not working. CSS Fonts 4, optical sizing and precedence supplies this context; this paper reports no comparison of browser engines.

```
/* After loading a suitable font with an opsz axis. */
.prose { font-optical-sizing: auto; }
```

For separate families, the strongest alternative is often an explicit editorial assignment. This needs no JavaScript to select the family and communicates the intended role directly. Font loading and licensing remain the author’s responsibility in either approach.

```
.caption { font-family: "PT Serif Caption", serif; }
.body { font-family: "PT Serif", serif; }
```

Media and container queries can pair family changes with layout rules when those rules define the design. CSS can switch families. The narrower question is whether that switch must follow an element's computed size as independent styles change. A fallback stack such as Text, Caption, serif gives a priority order, not an instruction to choose Caption at a smaller size.

> ### Font supplies opsz
>
> Start with browser optical sizing when its policy fits.

> ### Roles or layout define the choice
>
> Assign the separate families directly in CSS.

> ### Choice follows computed size
>
> Publish and execute an explicit family-selection policy.

A proposed decision order. Prefer the simplest mechanism that expresses the actual requirement.

Separate names need an authored relationship

Family names, design intent, optical metadata and file packaging are related but distinct. A variable font need not have an opsz axis. Static optical variants can carry STAT information; the older OpenType size feature also described optical design-size information and has been superseded by STAT. These are reasons to avoid a story in which optical selection began with this library or with variable fonts.

The OpenType opsz specification defines a point-based axis and discusses application policy. That does not turn arbitrary separate CSS family names into a discoverable size-selection relationship. Nor does it establish a universal conversion from CSS size to what a reader sees. This implementation reads computed CSS font size; it does not measure physical display size or viewing distance.

Foundry guidance can explain which drawing suits which uses. Authors must still translate that guidance into the particular page's roles, loading setup, fallbacks and transitions. A proposal to publish that translation is an authoring recommendation, not evidence that customers would buy differently.

What the default example selects

The default pair is PT Serif Caption and PT Serif, not a three-cut Micro/Text/Display system. ParaType's publisher description, distributed by Google Fonts, identifies the caption styles as intended for small point sizes. That supports the choice of a genuine optical pair; it does not establish the demo's threshold.

The demo author supplies a boundary at 16 CSS px and a hysteresis setting of 1 CSS px. With no previous cut, the underlying band is Caption below 16 and Text from 16. During live selection, hysteresis retains the current cut around that boundary. These are example settings, not measured recommendations.

Compare holds a second paragraph in PT Serif at the same requested size. When both use Text, the null result is expected. When the managed paragraph selects Caption, its drawing and spacing can differ. A family change can reflow the paragraph; stable selection does not mean stable line breaks.

PT SERIF CAPTION

Small text and large text can use different drawings.

PT SERIF

Small text and large text can use different drawings.

Both real families rendered at 28 CSS px for inspection. These enlarged specimens show the drawings, not an optimal use size, measured reading advantage, or the live demo's current selection.

The alternate Cormorant family set is not the optical-size evidence for this argument. Cursor and Distance modes are also outside the claim. The default Caption/Text pair is sufficient to

show authored selection without inferring a physical-distance sensor or a reading outcome.

Hysteresis makes selection depend on the route

The live selector, startOpszStepper as used by the React demo, supplies the retained browser trace. It records the ordered inputs 32, 16, 15, 14, 16, 17 and 12 CSS px, with Compare enabled. The managed result is Text, Text, Text, Caption, Caption, Text and Caption respectively. The comparison stays PT Serif. The observation table and raw trace expose the sequence without converting it into a survey or a statistical benchmark.

Descending from Text, the demo retains Text at 15 and selects Caption at 14. Ascending from Caption, it retains Caption at 16 and selects Text at 17. For this configuration, the downward exit requires a size below 15, whereas the upward exit is at or above 17. The dead zone reduces switching near the boundary by remembering the active cut. It deliberately trades a simple size-only mapping for state-dependent selection.

The one-shot applyOpszStepper API intentionally uses the initial raw ranges without hysteresis. Applied at 15 CSS px it chooses Caption immediately, rather than retaining a prior Text selection. Use the live API when reproducing this history-dependent trace.

The 16 px result therefore cannot be explained by a boundary label alone. The previous state and hysteresis value belong in any published policy. Nothing in this observation calibrates the boundary or proves that either drawing is better at that size.

The live selector rechecks on observed element or parent resizes, window resize and selected attribute changes, including class and style. It does not continuously sample every computed-size change. Continuously animated sizes and stylesheet-only changes without an observed trigger need separate verification. That is a trigger-coverage limit, not a browser-animation failure measured by this trace.

The default example uses contiguous, non-overlapping bands. A reusable policy should also specify gaps, overlaps and restarts: source inspection shows a live gap retains an existing cut, an initial unmatched gap writes no cut, and overlaps use the first matching resolved range. Restarting the live watcher resets selection history. These cases are not measured by the default browser trace.

Historical context without a market story

Frank Grießhammer's Source Serif gets optical sizes, published by Adobe in March 2021, describes size-specific drawing and the intent of optical subfamilies. It provides a designer's account, not a threshold experiment for PT Serif. The retained page has conflicting footer metadata; the source ledger uses the headline and URL date context rather than inventing an exact date from the footer.

The Typophile optical-sizes discussion archived from 2013 records debate about automation, manual choices and CSS alternatives. The TypeDrawers naming discussion from 2021 includes John Hudson, Dave Crossland and other participants with differing positions. These discussions preserve disagreement. They do not establish present-day application support, a consensus on ranges or measured adoption.

Commercial catalogues can make optical groups visible, but a label does not establish the axes in downloaded binaries. This paper makes no pricing, prevalence, revenue or packaging-performance claim. No direct forum quotation is reproduced or read in the transcript.

Objections answered

WHY NOT JUST USE CSS?

Often, you should. Assign separate families to roles or switch them alongside layout rules when that captures the design. The computed-size selector is an option when those rules do not conveniently express the intended policy. The argument is that the policy should be explicit, not that every policy needs this implementation.

DOESN'T NATIVE OPTICAL SIZING ALREADY SOLVE THIS?

For a suitable opsz font, start there. This demo's separate-family switch is a different delivery case; its fixed-family Compare is not a native-axis comparison. No claim of superiority to native optical sizing follows from it.

DOES CHANGING TO CAPTION IMPROVE READING?

That is not established here. The publisher’s intended use and the visible difference between drawings are not substitutes for a reading study. The evidence establishes selection behavior under configured conditions. It does not establish accessibility conformance, improved legibility or an optimal boundary.

DOES THE TOOL TAKE OVER FONT DELIVERY?

No. The author supplies and loads the fonts, chooses the policy, handles licenses and fallbacks, and checks transitions. A computed family name alone cannot prove which font painted every glyph. Loading and reflow need separate attention.

Recommendations

1. Inspect the font’s capabilities first. Use native optical sizing when its available axis and policy suit the design.
2. Prefer ordinary CSS when roles or layout rules already determine the separate-family choice.
3. If selection follows computed size, document the family list, units, bounds, hysteresis and fallback. Treat the mapping as a project decision rather than a universal prescription.
4. Test the actual text around transitions in both directions. Check that the intended fonts load and that changed line breaks remain acceptable.
5. Publish the guidance with the webfont setup. Explain what the implementation does not infer or validate.

Open questions

What format would make a foundry’s guidance easiest for authors to adapt without presenting intended use as a calibrated universal range? Where should a reader’s preference override automatic choices? Which real layouts are simpler with roles, and which justify following computed size? Those questions remain open. This trace cannot answer them through a demand estimate or a performance chart.

Method and verification boundary

The primary sources listed below were opened on 6 November 2026 and rechecked on 8 November. They support the documentary context above. This paper adopts paraphrases only. Its numbers are either the demo's configured inputs and observed family outputs, or Overpunch's supplied biographical disclosure. There are no external market numbers.

The saved live-browser trace was collected on 6 November 2026 with the PT pair loaded, Compare on, and the default 1 CSS px hysteresis. Existing research code records ordered input changes and inline/computed family results. A separate demo-first review of the clarified local build reproduced the key downward and upward switches. That supports a local selection demonstration; it does not certify all released wrappers or the current deployed commit.

A separate Chromium check of the published 1.1.1 core loaded the downloaded ESM directly into ordinary DOM text using the demo fonts. It reproduced Text/Text/Text/Caption/Caption/Text at 32,16,15,14,16,17 CSS px, and selected Caption for one-shot application at 15. Stopping restored the original serif family without changing the text; a subsequent size change left that restored family in place. The downloaded core hash matched the integrity-checked release evidence, and loaded entries for both PT families were recorded. This is a bounded check of the released core, not released-wrapper parity, cross-engine behavior or reading outcomes.

The data page retains the original observation JSON and an offline reconciliation script. That script checks the saved outputs; it is not a fresh browser measurement. Initial fallback-font reads were excluded from the selected trace, and loaded font entries are retained in the raw evidence. No timed rehearsal, released-wrapper parity test, cross-engine comparison, reading study, physical-distance measurement or performance benchmark is reported.

Inspected delivery paths share the core through the React wrapper, Framer import and Webflow bundle. The demo exercises the React path. Inspection is not an end-to-end test of downloaded Framer or Webflow artifacts, and no such release claim is made. No core or wrapper code was changed for this paper.

Sources

- S1 — W3C CSS Working Group, CSS Fonts 4: optical sizing. Standards context and variation precedence; no engine-parity experiment.
- S2 — Microsoft OpenType, Optical size axis, displayed update date 29 May 2024. Axis scale, static variants and application policy.
- S3 — Frank Grießhammer / Adobe, Source Serif gets optical sizes, March 2021. Designer’s account; not calibration of this demo.
- S5 — ParaType, PT Serif publisher description via Google Fonts, undated file. Caption intent, not a 16 px recommendation.
- S6 — Typophile contributors, Optical sizes in OT, 2013 archived discussion. Historical disagreement, not current support data.
- S7 — TypeDrawers contributors, including John Hudson and Dave Crossland, Optical-size naming discussion, page 2, May–June 2021. Naming/range debate, paraphrased without quotations.
- S9 — Microsoft OpenType, Registered size feature. Historical metadata; superseded by STAT.
- D1 — Retained ordered selection trace and reconciliation, 6 November 2026. Computed-size family selection only; configured boundaries are not reading recommendations.
- D2 — Published opszStepper 1.1.1 registry record, independently checked in Chromium using the downloaded ESM. Bounded selection, one-shot and cleanup behavior only.
- Disclosure — Overpunch’s supplied professional history and Typetin development status. These are interests disclosed by the author, not independent research results.