ULTRATEXTGEN RESEARCH · STUDY 5 · UNICODE, EMOJI AND FONTS

How Fast Do New Unicode Characters and Emoji Show Up? A Browser and Font Rendering Study

We drew every Unicode character from 8.0 to 18.0 and every emoji sequence in three browser engines, four font sets, 45 historical emoji fonts and 25 Chrome versions, then checked each result. The font is almost always the missing link.

MAIN FINDING

A new character fails to show because no installed font has a glyph for it, not because of the browser. New emoji reach the newest Noto font a median of 55 days after release, new scripts take about two years, and two Chrome changes (versions 131 and 142) fixed emoji that no font update could. The scope is Chromium, Firefox and WebKit on Linux. It is not a share of devices.

- Chromium on Linux complete: 59,852 code points, 5,244 emoji sequences, four font sets
- 45 historical emoji fonts and 25 Chrome versions complete
- Firefox and WebKit on Linux compared with Chromium
- Windows, macOS, iOS and Android not measured, device reports wanted
- Independent human review of the detector, raters wanted
- Real-device check of our missing-glyph test, device reports wanted

PUBLISHED
October 13, 2026

MEASURED
October 12 to 13, 2026, Chrome 155, Firefox 155, WebKit 26.6

VERSION
1.1 (corrected October 16, 2026)

EVIDENCE
Rendered pixels, font files, browser builds, vendor release pages

Canonical URL: https://ultratextgen.com/research/unicode-rendering-support-study/
UltraTextGen Research · Version 1.0 · Published October 13, 2026

SUMMARY

What the study found

A new Unicode character shows up on a screen only after a font has a glyph for it and the browser can put the pieces together. In the setups we tested, the font is almost always the slow link. Emoji reach fonts within weeks. New scripts take years. Browsers rarely get in the way, and when one did (Chrome before version 131 and again before version 142) we found the Chromium change that fixed it. We measured Chromium, Firefox and WebKit on Linux. We did not measure Windows, macOS, iOS or Android, so none of this is a share of devices.

55 days

median wait from an Emoji release to the first Noto font file that draws it

Twelve Emoji releases, 5.0 to 18.0. The middle half ran from 3 to 94 days, the extremes from a font that came 32 days before the release to a wait of 273 days. The six releases from 14.0 on had a median of 12.5 days, the six before them 106.5. Every day from 2017 to 2026, the newest font drew on average 93% of the emoji already standardized, and the longest stretch below 100% lasted 273 days.

MEASURED FROM FONT FILES

0 of 19

Emoji 18.0 sequences drawn correctly with Ubuntu 24.04’s own fonts

With the newest Noto fonts it is 19 of 19. For Emoji 17.0 (13 months old) the Ubuntu fonts draw 15 of 163 (9.2%) and the newest Noto 163 of 163. A reader with a current font set sees a different web than one with the packaged fonts, on the same browser.

RENDERED IN CHROMIUM 155 LINUX FONTS ONLY

2

Chrome changes that decided whether the same font worked

Up to Chrome 130, the red heart and 83 other text-default emoji followed by the emoji selector are drawn as black symbols when the page lists a text font first, and from Chrome 131 all are drawn in colour. Chrome 113 to 141 splits 60 two-tone “people wrestling” emoji into pieces and Chrome 142 draws them whole. Bisecting Chromium builds narrowed each to a few commits. In each the only emoji change is one that makes the browser treat a variation selector or a not-yet-known emoji code point as emoji.

TESTED ON 25 RELEASES AND 25 BUILDS · CHROMIUM-SPECIFIC

0.01% to 99.99%

how many missing glyphs our own site’s device check could see, before and after we fixed it

The check compared each character with a “guaranteed unassigned” reference that is actually an invisible character, so a drawn box never matched it. Of about 47,700 characters a font set could not draw, the old check flagged 4. A real unassigned reference flags all but 4, and a second reference flags every one, with no false alarms, on three of four font sets in Chromium and WebKit. In Firefox it flags none, because Firefox draws a different box for every missing character. It is calibrated on Linux only, and real-device reports are wanted.

RUN ON 68,733 CODE POINTS · NOT YET ON REAL DEVICES

NOT TESTED **What this study cannot tell you**

It cannot tell you what share of people can see a character, or what any phone, Windows PC or Mac shows. Rendering here means Chrome 155, Firefox 155 or WebKit 26.6 on Ubuntu 24.04 with a named font set. It does not cover keyboards and input, which are a separate step, or text in browser menus and address bars. The detector was checked by one reader, not by independent reviewers. UltraTextGen makes text tools whose output depends on these fonts, which is why we ran this. See what we are asking for and read the limitations.

WHAT TO DO

What this means for you

Sharing a brand-new emoji or symbol? Expect many readers to see a box for weeks to months. On the Linux fonts we tested, characters from Unicode 16.0 to 18.0 (all under two and a half years old) were drawn

0% to 0.6% of the time with the packaged fonts and 1.6% to 8.7% with the newest Noto fonts. For Windows and iPhone we only have the dates the vendors say they shipped each Emoji release (see the table below): about six months after the release for iPhone and one to one and a half years for Windows 11.

Seeing boxes? The box is drawn by the device that shows the text. Our gallery shows what each kind of failure looks like and which one you have. The practical guide is why text shows as boxes, and for a name in a game see why your name shows as boxes.

Running a website? Put the colour emoji family before the symbol families in your font stack. With the newest fonts, the same text drew 100% correctly with the emoji family first and 94.9% with it last, because 78 emoji turned into black symbols. In Chrome 130 and earlier, a text font listed before the emoji font also draws the red heart as a black symbol even when the page asks for the emoji form.

Choosing a font for a new script or symbol? A pooled “percent supported” figure hides the big blocks. Of the 8,319 characters this site offers, 14 are not drawn even with the newest Noto fonts. They include the three new currency signs (rufiyaa, dirham, Omani rial).

Building text tools? Detect a missing glyph by drawing it and comparing it with a real unassigned character, and keep the check honest by asking users for both outcomes. We broke this ourselves, and the fix and its limits are in what we found on our own site.

DESIGN

Questions and design

A character exists the day Unicode standardizes it. You can see it only if every link in a chain holds: a font contains a glyph, the system exposes the font, the browser picks it, the shaping engine assembles sequences, and the pixels are drawn. We asked three questions. How long after standardization does a character become drawable? Which link breaks when it does not? Does a text tool offer characters that exist but may not draw?

We built two datasets from the Unicode Consortium’s own files and never pooled them. Dataset 1 is 59,852 code points of Unicode 8.0 to 18.0, of which 44,636 are the study range 10.0 to 18.0. Dataset 2 is 5,244 emoji sequence lines, of which 1,579 are fully-qualified study sequences. A deterministic harness draws every item in a fixed cell and records the pixels, the font the browser reports through its developer protocol, the font’s own character map, the advance width and the glyph count. A detector turns those signals into one class per item. We checked the detector by eye on two blind samples (see Method).

Four font sets, five layers

CONDITION	WHAT IS INSTALLED
STOCK CONTAINER	Ubuntu 24.04 container fonts as found, including GNU Unifont, a font that draws a hex-code box for every missing character

CONDITION	WHAT IS INSTALLED
MINIMAL	The stock set without Unifont and Inter: a typical server image with DejaVu, Liberation, FreeFont, WenQuanYi, IPA and Noto Color Emoji 2.047
UBUNTU PACKAGED	Minimal plus the Noto font packages Ubuntu 24.04 ships (core, extra, mono, UI, dated 2020 and 2023; CJK dated 2023)
NEWEST NOTO	The newest Regular file of every Noto family, Noto CJK, and the newest Noto Color Emoji (2.057)

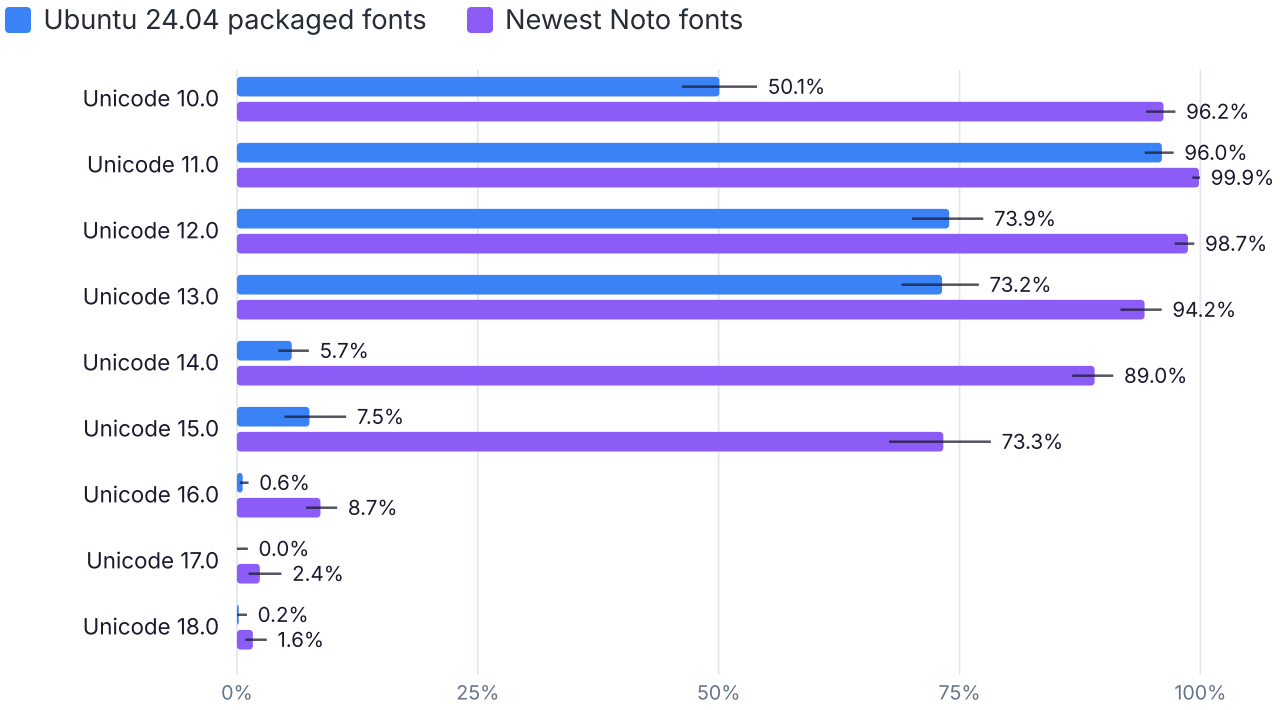
The layers: (A) every item in the four sets; (B) font support lag, from the dates of 45 committed Noto Color Emoji files and 45 monthly Noto releases; (C) a historical replay that holds Chromium 155 fixed and swaps in each of the 45 emoji fonts; (D) a browser matrix of Chrome 113, 120, 128, 135, 141 and 155 against eight emoji fonts, plus a check of every Chrome release between the versions that differ, and two bisects over 25 Chromium snapshot builds; (E) other engines and font formats: Firefox and WebKit, and the Windows-compatible, COLRv1 and Twemoji COLR emoji fonts. The browser is Chrome for Testing 155.0.8059.39, headless, with fixed flags, viewport and scale.

RESULTS

Single characters: coverage falls in steps, not smoothly

For code points from Unicode 18.0 (26 days old at measurement), 1 of 547 scripts and symbols were drawn as intended with the Ubuntu packaged fonts, and 9 of 547 (1.6%) with the newest Noto. For Unicode 16.0 the figures are 7 of 1,189 (0.6%) and 103 of 1,189 (8.7%). For Unicode 15.0 and 15.1 together they are 7.4% and 72.0%.

Figure 1. Share of code points drawn as intended, by Unicode release. Chromium 155 on Linux, with the fonts Ubuntu 24.04 packages and with the newest Noto fonts. Whiskers are 95% intervals.



▼ **Data behind this chart**

RELEASE	PACKAGED FONTS: DRAWN	PACKAGED FONTS: SHARE	NEWEST NOTO: DRAWN	NEWEST NOTO: SHARE
UNICODE 10.0	314 of 627	50.1%	603 of 627	96.2%
UNICODE 11.0	647 of 674	96.0%	673 of 674	99.9%
UNICODE 12.0	397 of 537	73.9%	530 of 537	98.7%
UNICODE 13.0	341 of 466	73.2%	439 of 466	94.2%
UNICODE 14.0	47 of 828	5.7%	737 of 828	89.0%
UNICODE 15.0	20 of 266	7.5%	195 of 266	73.3%

RELEASE	PACKAGED FONTS: DRAWN	PACKAGED FONTS: SHARE	NEWEST NOTO: DRAWN	NEWEST NOTO: SHARE
UNICODE 16.0	7 of 1,189	0.6%	103 of 1,189	8.7%
UNICODE 17.0	0 of 337	0.0%	8 of 337	2.4%
UNICODE 18.0	1 of 547	0.2%	9 of 547	1.6%

Code points of scripts and symbols only. Han, Seal, Tangut, Egyptian hieroglyphs, Jurchen, Anatolian hieroglyphs, Khitan, Nushu and Cuneiform are left out because one huge block can decide a release's total (87% of Unicode 18.0 is Seal). Unicode 12.1 and 15.1 are too small to chart.

The curve is a map of font build dates. With Ubuntu's fonts coverage is 73% for Unicode 13.0 and 6% for 14.0. That cliff is the date of Ubuntu's Noto snapshot (December 25, 2020), not a property of the characters. Coverage by age therefore has nine distinct ages, one per release, so "age" and "release" cannot be separated in this data.

When one block decides the total. Han (27,389), Seal (11,328), Tangut (7,061), Egyptian hieroglyphs (4,034) and Jurchen (965) are 85% of Dataset 1. Unicode 18.0 is 87% Seal, 11,328 of 13,007 code points, and no font in any condition has a glyph for Seal. The pooled share for the newest releases is near 0% under every condition for that reason alone.

Single-character failure is font coverage. Of the code points that were not drawn as intended with the packaged fonts, 99.96% had no installed font that maps them. A box can also come from a font that maps a character but draws nothing (6 code points) or from a pan-Unicode placeholder font (stock set: 7,220 hex-code boxes). A replacement symbol never appeared on its own.

Newest Noto closes much of the gap, not all. It rescued 57.5% of the code points that failed in the minimal set and 41.2% of those that failed with the packaged fonts. It cannot help where no Noto font exists: 2,039 non-logographic code points are not drawn as intended in any of the four font sets, and the three Unicode 18.0 currency signs (U+20C2, U+20C3, U+20C4) are drawn by none.

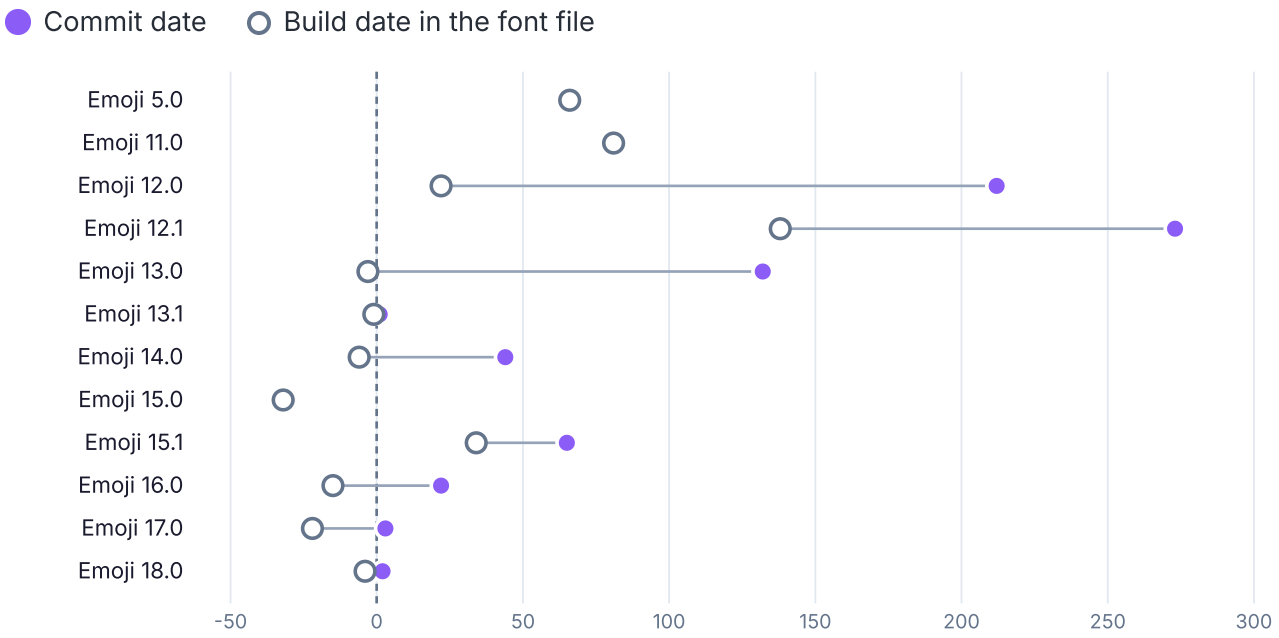
RESULTS

Emoji: weeks to reach a font, then a long tail to reach you

We shaped every emoji sequence with each of the 45 Noto Color Emoji files committed between March 2017 and September 2026, using HarfBuzz, and recorded the first file that composes it. Each Emoji release

arrives in one file, so the lag is a batch value. Across 12 releases the median is 54.5 days after the release, with a middle half of 2.8 to 93.8 days and a 90th percentile of 204. The series is short and noisy, but the recent releases are faster.

Figure 2. Days from an Emoji release to the first Noto Color Emoji binary that composes its sequences. Filled dot: date the binary was committed to the font's repository. Ring: date stamped inside the font file. Negative means the font came first.



▼ **Data behind this chart**

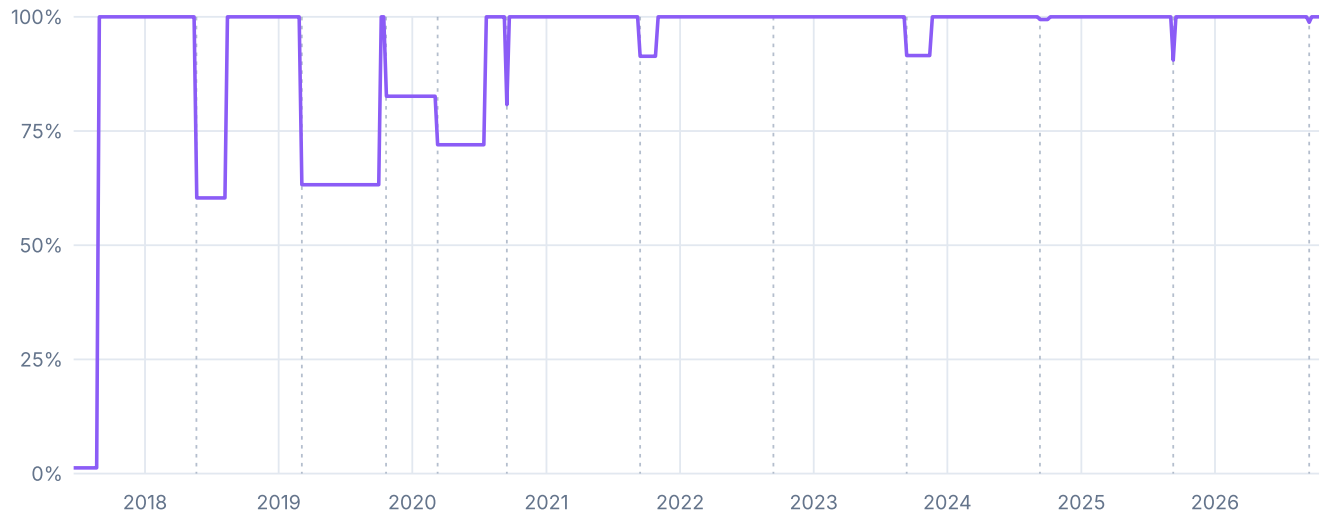
RELEASE	RELEASED	LAG, COMMIT DATE (DAYS)	LAG, BUILD DATE (DAYS)
EMOJI 5.0	2017-06-20	66	66
EMOJI 11.0	2018-05-21	81	81
EMOJI 12.0	2019-03-05	212	22
EMOJI 12.1	2019-10-21	273	138
EMOJI 13.0	2020-03-10	132	-3
EMOJI 13.1	2020-09-15	1	-1
EMOJI 14.0	2021-09-14	44	-6
EMOJI 15.0	2022-09-13	-32	-32
EMOJI 15.1	2023-09-12	65	34
EMOJI 16.0	2024-09-10	22	-15

RELEASE	RELEASED	LAG, COMMIT DATE (DAYS)	LAG, BUILD DATE (DAYS)
EMOJI 17.0	2025-09-09	3	-22
EMOJI 18.0	2026-09-15	2	-4

The two dates differ by months for some releases because binaries were committed long after they were built (the Emoji 12.0 font was built in March 2019 and committed in October 2019). Every item of a release first appears in one binary, so these are 12 batch values, not a sample of 1,559 independent items.

A release leaves a gap that stays open until the next font file. Replaying day by day from June 2017 to October 2026, the newest committed font drew on average 92.8% of the emoji already standardized. It drew all of them on 77.4% of days, fewer than 90% on 18.6% of days, and the longest stretch below 100% was 273 days.

Figure 3. Share of already-standardized emoji (Emoji 5.0 onward) that the newest committed Noto Color Emoji binary can compose, every day from 2017-06-20 to 2026-10-12. Each vertical line is an Emoji release.



▼ **Data behind this chart**

YEAR	MEAN SHARE COMPOSED	LOWEST DAY	HIGHEST DAY
2017	66.6%	1.3%	100.0%
2018	91.2%	60.4%	100.0%
2019	75.2%	63.3%	100.0%
2020	86.6%	72.0%	100.0%
2021	99.0%	91.4%	100.0%
2022	100.0%	100.0%	100.0%
2023	98.5%	91.5%	100.0%
2024	100.0%	99.4%	100.0%
2025	99.9%	90.5%	100.0%
2026	100.0%	98.8%	100.0%

This is the best case for a reader: it assumes the reader has the newest committed font on the day it is committed. Series drawn weekly; the table is computed from the daily file. Emoji 5.0 releases the first

day of the series.

When vendors say they shipped

The font file is the first step. The device vendor is the next. We did not measure Windows or iPhone rendering. We read each vendor's own release pages for the first date they list an Emoji version, and counted days from the Emoji release. Apple does not name the Emoji version, so those rows are inferred from the emoji it lists. Where a vendor lists several steps we used the first one it names that is not an early test build: Microsoft's Release Preview channel for Windows 16.0 and 17.0, and the general update for 15.0.

WHO	EMOJI VERSION	RELEASED	FIRST SHIPPED	DAYS	WHAT THE PAGE SAYS
NOTO COLOR EMOJI FONT	Emoji 16.0	2024-09-10	2024-10-02	22	font file committed (this study)
NOTO COLOR EMOJI FONT	Emoji 17.0	2025-09-09	2025-09-12	3	font file committed (this study)
WINDOWS 11	Emoji 15.0	2022-09-13	2023-11-14	427	KB5032190, "adds support for Unicode Emoji 15"
WINDOWS 11	Emoji 16.0	2024-09-10	2026-02-17	525	Release Preview, then KB5077241 optional preview on 2026-02-24
WINDOWS 11	Emoji 17.0	2025-09-09	2026-09-10	366	Release Preview KB5124010; Microsoft lists an optional preview on 2026-09-22
IPHONE (IOS)	Emoji 15.1	2023-09-12	2024-03-05	175	iOS 17.4, version inferred from the emoji listed
IPHONE (IOS)	Emoji 16.0	2024-09-10	2025-03-31	202	iOS 18.4, version inferred from the emoji listed
IPHONE (IOS)	Emoji 17.0	2025-09-09	2026-03-24	196	iOS 26.4, version inferred from the emoji listed

Dates are from Microsoft Learn, the Windows Insider blog, Microsoft Support and Apple Support release pages, listed under Sources. Windows 10 has no listed Emoji version after 12.0, and Microsoft's support pages place Emoji 15.1 only under Windows 11. We found no primary page that gives an Emoji version for

any Android or Samsung release, or an Emoji 18.0 date for Apple or Microsoft. Absence on a page is not proof of absence elsewhere. For the current state of one release see where Unicode 17.0's emoji work right now.

RESULTS

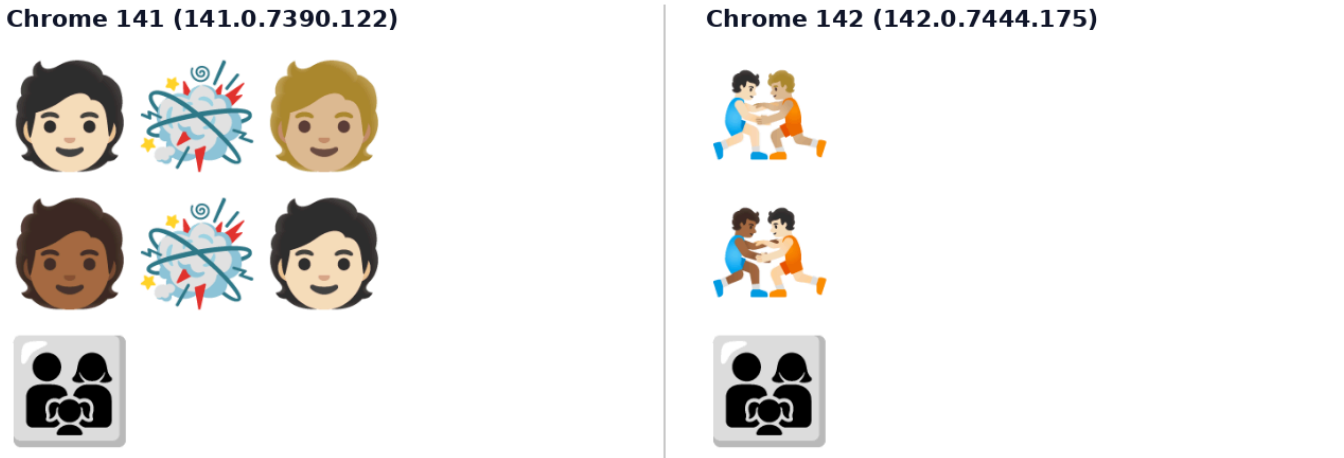
When the font is not the problem: two Chrome changes

For single characters the browser never mattered: the font either had a glyph or it did not. For emoji sequences it mattered twice. We swapped Chrome 113, 120, 128, 135, 141 and 155 against eight historical emoji fonts and compared every cell, then tested every release in between where cells differed. The six browsers agreed except for two families of sequences. Each has a release that fixed it and, for both, we found the Chromium change that most likely did.

Chrome 142: a new emoji inside a sequence

60 sequences from Emoji 17.0, the two-skin-tone "people wrestling". With Noto Color Emoji 2.051, Chrome 113 to 141 drew them as separate pieces and Chrome 142 and later drew all 60 as one picture.

Figure 4. The same font file and the same page in Chrome 141 and Chrome 142. Row 3 is a control that works in both.



Same font file (Noto Color Emoji 2.051) and same page. Rows 1 and 2: two-tone "people wrestling" (Emoji 17.0). Row 3: a family, a control that works in both

EMOJI FONT COMMITTED	CHROME 113	CHROME 120	CHROME 128	CHROME 135	CHROME 141	CHROME 155
AUG 2018	26.2%	26.2%	26.2%	26.4%	26.4%	26.4%

EMOJI FONT COMMITTED	CHROME 113	CHROME 120	CHROME 128	CHROME 135	CHROME 141	CHROME 155
OCT 2019	42.7%	42.7%	42.7%	42.8%	42.8%	42.8%
SEP 2020	72.5%	72.5%	72.5%	72.7%	72.7%	72.7%
SEP 2022	81.2%	81.2%	81.2%	81.4%	81.4%	81.4%
NOV 2023	88.7%	88.7%	88.7%	88.9%	88.9%	88.9%
OCT 2024	89.2%	89.2%	89.2%	89.4%	89.4%	89.4%
SEP 2025 (V2.051)	94.8%	94.8%	94.8%	95.0%	95.0%	98.8%
SEP 2026 (V2.057)	96.0%	96.0%	96.0%	96.2%	96.2%	100.0%

Share of the 1,579 fully-qualified study sequences that compose, by emoji font file (listed by commit date) and Chrome release, with DejaVu Sans before the emoji font in the page. Chrome 141 and 155 differ only for the fonts that contain the 60 sequences (Sep 2025 and Sep 2026). The 128 to 135 step in every row is the second change, below.

We tested every Chrome release from 142 to 154: all compose the 60. Then we bisected Chromium's own snapshot builds between the 141 and 142 branch points (13 builds, positions 1509326 to 1522578). The last build that fails is position 1516483 and the first that composes is 1516493, ten commits apart, all from September 17, 2025.

CHROMIUM SNAPSHOT POSITION	VERSION STRING	TWO-TONE PEOPLE WRESTLING COMPOSED
1509326	141.0.7390.0	0 of 60
1515958	142.0.7418.0	0 of 60
1516379	142.0.7419.0	0 of 60
1516483	142.0.7419.0	0 of 60
1516493	142.0.7419.0	60 of 60
1516500	142.0.7419.0	60 of 60
1516516	142.0.7419.0	60 of 60
1516543	142.0.7419.0	60 of 60

CHROMIUM SNAPSHOT POSITION	VERSION STRING	TWO-TONE PEOPLE WRESTLING COMPOSED
1516588	142.0.7420.0	60 of 60
1516792	142.0.7420.0	60 of 60
1517619	142.0.7422.0	60 of 60
1519273	142.0.7431.0	60 of 60
1522578	142.0.7444.0	60 of 60

One of those ten commits changes how the browser decides where an emoji run starts and stops. Its title is “[icu] Include reserved ranges for Emoji to GetEmojiSegmentationCategory” (Chromium commit `fb172d8c5b7d`, position 1516492). Before it, the splitter counted a code point as emoji only if the browser’s Unicode data listed it as one. After it, a code point that is still unassigned in that data but lies in a range reserved for pictographs also counts, so a new emoji can join a sequence before the data catches up. The sequences use U+1FAEF (fight cloud), added in Unicode 17.0, and the commit’s own unit test names it.

We measured what each build’s Unicode data knew by asking it whether a code point is unassigned. The data is older than you might expect.

BROWSER	KNOWS A UNICODE 16 EMOJI (U+1FAE9)	KNOWS A UNICODE 17 EMOJI (U+1FAEF)	KNOWS A UNICODE 17 CURRENCY SIGN (U+20C1)	KNOWS A UNICODE 18 CURRENCY SIGN (U+20C2)
CHROME 113 TO 142	no	no	no	no
CHROME 143 TO 147	yes	no	no	no
CHROME 148 TO 155	yes	yes	yes	no
FIREFOX 155	yes	yes	yes	no
WEBKIT 26.6	yes	yes	yes	no

Chrome 142 shipped data from Unicode 15.1, so it treated U+1FAEF as unassigned, and the change above is what let it compose the sequence anyway. Chrome 148 is the first build that knows Unicode 17. None of the

19 builds, Firefox or WebKit knows Unicode 18, yet Emoji 18.0 sequences compose in Chrome 155 with the newest font.

Chrome 131: the emoji selector

A character such as the red heart has a text form and an emoji form, and U+FE0F asks for the emoji form. We drew 207 such sequences with a stack that lists a text font with the glyph (DejaVu Sans) before the emoji font, which is how many websites are written. In Chrome 113, 120 and 128, 84 of them (the red heart, the smiling face, the warning sign and similar) were drawn by the text font as black symbols, with the selector ignored. In Chrome 135, 141 and 155 all 207 were drawn in colour. A direct test of 12 of them across 13 releases places the step exactly: 0 of 12 in colour up to Chrome 130 and 12 of 12 from 131.

CHROME RELEASE	DRAWN IN COLOUR, OF 12 TEST SEQUENCES
113.0.5672.63	0 of 12
120.0.6099.109	0 of 12
128.0.6613.137	0 of 12
129.0.6668.100	0 of 12
130.0.6723.116	0 of 12
131.0.6778.264	12 of 12
132.0.6834.159	12 of 12
133.0.6943.141	12 of 12
134.0.6998.165	12 of 12
135.0.7049.114	12 of 12
141.0.7390.122	12 of 12
142.0.7444.175	12 of 12
155.0.8059.39	12 of 12

We bisected Chromium snapshots between the 130 and 131 branch points (positions 1356013 to 1368529; 12 builds). The last build that draws black symbols is position 1359599 and the first that draws colour is 1359605, six commits apart, all from September 24, 2024.

CHROMIUM SNAPSHOT POSITION	VERSION STRING	DRAWN IN COLOUR, OF 12
1356035	130.0.6723.0	0 of 12
1359161	131.0.6737.0	0 of 12
1359563	131.0.6738.0	0 of 12
1359599	131.0.6738.0	0 of 12
1359605	131.0.6738.0	12 of 12
1359618	131.0.6738.0	12 of 12
1359668	131.0.6738.0	12 of 12
1359758	131.0.6739.0	12 of 12
1359946	131.0.6740.0	12 of 12
1360727	131.0.6742.0	12 of 12
1362283	131.0.6751.0	12 of 12
1368524	131.0.6778.0	12 of 12

Only one of the six concerns emoji. Its title is “Move FontVariantEmoji and FontVariationSequences to stable” (commit `1d107b950528`, position 1359605, September 24, 2024). It switches on two features at once: the CSS property that chooses text or emoji presentation, and the handling of Unicode variation sequences, which is what makes the browser take U+FE0F into account when it picks a font. The first snapshot that draws colour is exactly that position.

What this shows, and what it does not. Each bisect isolates a window of commits, and in each the change we name is the only one that touches emoji or text handling. Its message and the behavior we measure fit. We did not rebuild Chromium with and without either change, so the attribution rests on that fit and not on a controlled build. The earlier hypothesis in this study, that very long sequences hit a shaping limit, was wrong: the 60 sequences have seven code points, and other sequences of up to ten code points compose in Chrome 113 with the same font. The second effect needs a stack with a text font before the emoji font: put the emoji family first and the same text draws in colour in every Chrome we tested.

RESULTS

The CSS font order changes the outcome

We kept the fonts and the browser fixed and changed only the order of font families in the CSS. With the newest Noto fonts, 100% of the 1,579 fully-qualified sequences composed when the colour emoji family came before the symbol families. With the symbol families first it was 94.9%: 78 emoji turned into black symbols and 2 split into pieces. With the Ubuntu fonts and the emoji family last (the common default), 89.4% composed in every order, because the font file predates Emoji 17.0 and the rest are the newer releases. There the order only changed how the failures looked. With Ubuntu's fonts, moving the emoji family first turned 86 boxes into 16 boxes plus 70 half-drawn sequences, and the stock container went from 16 boxes, 70 partial and 81 split sequences to 151 split and 16 monochrome. The stack decides how a failure looks and, for symbols first, adds failures the fonts alone would not have.

FONT SET AND ORDER	COMPOSED	BOXES	SPLIT INTO PIECES	PART BOXES
NEWEST NOTO, EMOJI FAMILY FIRST	100.0%	0	0	0
NEWEST NOTO, SYMBOL FAMILIES FIRST	94.9%	0	2	0
UBUNTU PACKAGED, EMOJI FAMILY LAST	89.4%	86	81	0
UBUNTU PACKAGED, EMOJI FAMILY FIRST	89.4%	16	81	70
STOCK CONTAINER, EMOJI FAMILY LAST	89.4%	16	81	70
STOCK CONTAINER, EMOJI FAMILY FIRST	89.4%	0	151	0

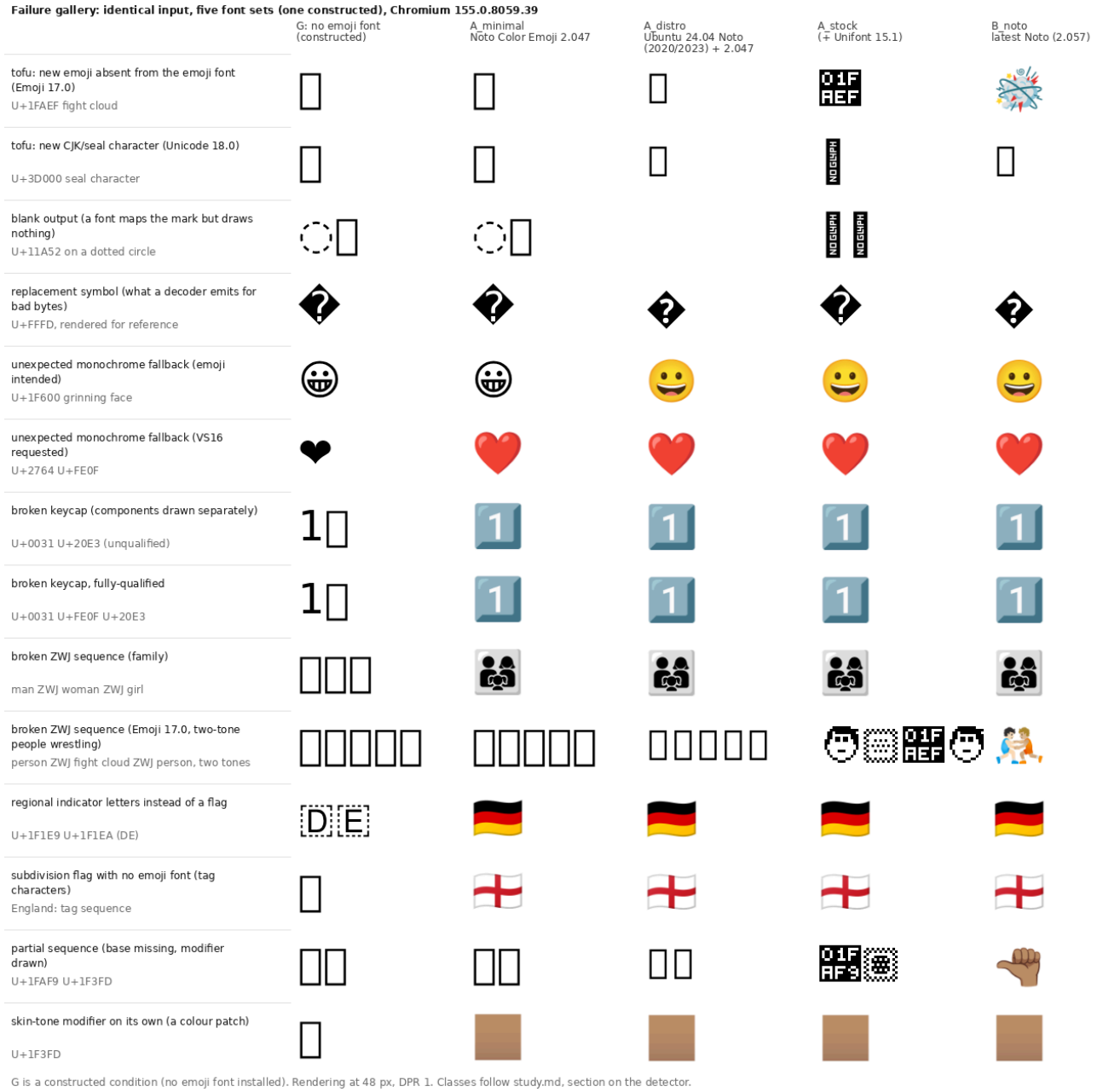
The stock container draws a hex-code box for a missing character, so nothing looks blank, but it is not the character.

RESULTS

What each failure looks like

The same input drawn under five font sets. Column 1 is a constructed set with no emoji font, which stands in for a deleted or missing emoji font. It contributes to no statistic. Failure comes in distinct types, and the type tells you the cause: a box is a missing glyph, an unbroken run of boxes is one missing piece pulling a whole sequence into one font, and pieces side by side mean the font has the parts but not the combination.

Figure 5. One input per row, five font sets in Chromium 155. Rendered at 48 px.



RESULTS

Other browser engines and other emoji fonts

The main run uses Chromium. We repeated the draw in WebKit 26.6 and Firefox 155 (the builds Playwright ships), on Linux, using the same fonts through the same font configuration. A character counts as drawn if its picture differs from the box the engine draws for an unassigned character. A sequence counts as

composed if it is no wider than one emoji. We compared each engine with this study's Chromium results for every item.

ENGINE AND FONT SET	CODE POINTS COMPARED	SAME VERDICT AS CHROMIUM	DRAWN ONLY BY ENGINE / ONLY BY CHROMIUM	SEQUENCES COMPARED	SAME VERDICT	DRAWN ONLY BY ENGINE / ONLY BY CHROMIUM
WEBKIT 26.6, MINIMAL	57,916	99.99%	0 / 3	3,963	99.97%	1 / 0
WEBKIT 26.6, UBUNTU PACKAGED	57,915	99.99%	0 / 3	3,963	100.0%	0 / 0
WEBKIT 26.6, NEWEST NOTO	57,913	99.99%	0 / 5	3,963	100.0%	0 / 0
FIREFOX 155, MINIMAL	57,916	100.0%	0 / 1	3,963	99.72%	1 / 10
FIREFOX 155, UBUNTU PACKAGED	57,915	100.0%	0 / 1	3,963	99.75%	0 / 10
FIREFOX 155, NEWEST NOTO	57,913	100.0%	0 / 0	3,963	100.0%	0 / 0

Right-to-left characters (716 code points) are left out of the code point comparison because the test cell anchors text at its left edge, which WebKit draws outside the cell for those scripts. We confirmed by eye that those characters do draw. Chromium compared with its own earlier run agrees on 100% of code points in two font sets and 99.98% in the third (12 characters whose glyph looks like a box).

Both engines agree with Chromium on at least 99.7% of items in every font set, and on every sequence with the newest Noto fonts. The few differences are in which fallback font an engine finds, and none is about composition. WebKit does not draw 3 Old Italic letters (U+1032D to U+1032F) that Chromium and

Firefox draw, and 2 Sunuwar digits with the newest fonts. Firefox draws a box for one Unicode 18.0 modifier letter (U+208F) that Chromium draws from a fallback font in the minimal and packaged sets. In the Ubuntu packaged and minimal sets, Firefox draws the broken pair for 10 skin-tone sequences of the thumb sign emoji at a different width from Chromium, which our width test reads as composed in one engine and not in the other; no font in those sets can draw the base emoji at all. We found no case of an engine drawing a character or sequence correctly that another engine could not, given the same fonts.

Emoji font formats

The main run uses the bitmap (CBDT) build of Noto Color Emoji. We also drew everything with the vector colour build (COLRv1), the build made for Windows, and the Mozilla build of Twemoji. The three Noto builds of version 2.057 gave the same class for all 3,963 fully-qualified sequences, so for composition and coverage the file format makes no difference in Chromium on Linux. Twemoji (Emoji 15 era) composes the older releases, then falls behind: it has no combined glyph for most skin-tone versions of Emoji 4.0 and 13.1 and none for Emoji 15.1 onward.

RELEASE	NOTO BITMAP (CBDT)	NOTO VECTOR (COLRV1)	NOTO WINDOWS BUILD	TWEMOJI (MOZILLA COLR 0.7)
EMOJI 0.6	89.4%	89.4%	89.4%	87.9%
EMOJI 1.0	95.7%	95.7%	95.7%	95.7%
EMOJI 4.0	100.0%	100.0%	100.0%	30.9%
EMOJI 5.0	100.0%	100.0%	100.0%	58.2%
EMOJI 13.1	100.0%	100.0%	100.0%	5.5%
EMOJI 15.1	100.0%	100.0%	100.0%	0.0%
EMOJI 16.0	100.0%	100.0%	100.0%	0.0%
EMOJI 17.0	100.0%	100.0%	100.0%	0.0%
EMOJI 18.0	100.0%	100.0%	100.0%	0.0%

Share of fully-qualified sequences composed in Chromium 155 with that single emoji font plus DejaVu Sans. Noto is below 100% for the oldest releases because the page lists DejaVu Sans first and DejaVu has its own

black glyph for some emoji (the grinning faces, for example), the same effect as in the CSS section. We could not obtain the OpenMoji font from a package source, so it is not tested.

RESULTS

Combining marks on a real letter

Combining marks (accents and other marks that attach to a letter) cannot stand alone, so the main run draws each on a dotted circle. Chromium picks one font per cluster, so a mark could fail there only because the dotted circle pulled in a different font. We redrew every mark that failed, on a real letter of its own script (the Latin letter a for shared marks). Of the 398 marks that failed in the stock container and map to an installed font, 276 drew correctly on a real letter. With the Ubuntu packaged fonts it was 15 of 17, and in the minimal set 1 of 7. Counting every mark that failed on the circle, 577 of 710 in the stock container drew on a real letter. The dotted-circle test overstates mark failures, so the numbers for marks in the main run are upper bounds on failure.

OUR OWN SITE

What we found on our own site

UltraTextGen offers 8,319 characters and emoji for copying or generates them from its text styles. We drew every one in the same four font sets. With the Ubuntu packaged fonts 99.4% draw as intended and 39 do not. With the newest Noto fonts 14 do not. The ones that fail even there include the three new currency signs, rufiyaa, dirham and Omani rial, which are in Unicode 18.0 and have no Noto font yet. In the minimal font set 16.1% of the characters fail, because they come from a long tail of scripts that need a font install. Of the site’s text styles, only Ultra Cherokee produces a failing character with the Ubuntu packaged fonts.

FONT SET	CHARACTERS TESTED	DRAWN AS INTENDED	NOT DRAWN
MINIMAL	8,285	82.61%	1336
STOCK CONTAINER	8,290	86.48%	1113
UBUNTU PACKAGED	8,282	99.44%	39
NEWEST NOTO	8,287	99.75%	14

The device check could not see a missing glyph

Pages that show a character list carry a “may not show on your device” notice, driven by a small script that draws a test character on a canvas and compares it with a reference that no font should map. The

reference was U+E01ED, described in the code as unassigned. It is Variation Selector-254, which is default-ignorable and draws nothing. A box is never equal to a blank bitmap, so every missing glyph passed as supported. A second defect hid the first: the character the script sampled from a style could be half of a surrogate pair, so fixing only the reference would have started flagging 24 styles that draw fine.

We ran the script's own code against about 67,000 code points in each font set. It flagged 4 of 47,696 unsupported characters with the Ubuntu fonts. Replacing the reference with a real unassigned code point (U+0378) detects 99.99%, and adding the noncharacter U+FFFF reaches 100% in three font sets, with no false alarm in any. The stock container is the exception: Unifont draws a different box for every missing character, so no equality test can see them, and combining references reaches 87%. All of this is Chromium; the next section says what happens elsewhere.

FONT SET	UNSUPPORTED CHARACTERS	OLD CHECK FLAGGED	ONE REAL REFERENCE FLAGGED	TWO REFERENCES AND BLANK FLAGGED
UBUNTU PACKAGED	47,696	0.01%	99.99%	100.0%
MINIMAL	59,754	0.01%	99.99%	100.0%
NEWEST NOTO	45,791	0.0%	100.0%	100.0%
STOCK CONTAINER (HEX-CODE BOXES)	54,532	0.01%	0.0%	86.83%

Non-emoji code points from this study's dataset and the site's inventory. Truth is this study's own classification of the same characters. There were no false alarms in any font set or variant.

The corrected check in three engines

We ran the corrected check (two references, U+0378 and U+FFFF) in Chromium, Firefox and WebKit on every code point, in the same font sets. In Chromium and WebKit it flags every missing glyph in three of the four font sets. In Firefox it flags none, with no false alarms: Firefox draws a different hex-code box for every missing character, as the stock container does, and a test that asks whether two drawings are identical cannot see that. We tried a variant that also flags a closed frame with the same outer size as a reference box. It found 90% to 100% in three Firefox font sets and 0.2% in the fourth, and its false alarms depended on what the page had drawn before: 5 of 11,380 in one run and 1,185 to 1,382 in four fresh runs of the same font set, because Firefox's canvas text for a character changed between draws. We did not ship it. In Firefox the notice therefore still never appears, which is what it did before the fix, so nothing got worse, and the device test below is how we will learn what Firefox needs.

ENGINE	UBUNTU PACKAGED	MINIMAL	NEWEST NOTO	STOCK CONTAINER
CHROMIUM 155	100.0% (0)	100.0% (0)	100.0% (0)	86.5% (0)
WEBKIT 26.6	100.0% (0)	100.0% (0)	100.0% (0)	86.5% (0)
FIREFOX 155	0.0% (0)	0.0% (0)	0.0% (0)	0.0% (0)

Share of the characters a font set cannot draw that the corrected check flags as missing, with false alarms (characters it flagged that the font set can draw) in brackets. Truth is this study's classification of the same characters; the check draws on a canvas with a generic sans-serif font, so a few differences come from that font choice.

What we changed. The check now compares against U+0378 and U+FFFF and samples a real character from each style. We ran it in a browser against the site's category page for ancient fonts: it flags Ultra Tibetan and Ultra Yi on the minimal fonts, flags nothing with full Noto, and raised no errors. **What we cannot yet say.** This is calibrated on Linux and on three engines. Other systems draw other boxes, and some draw nothing for an unmapped character, so the check could miss a missing glyph or flag a good one on a device we have not tried. That is why the device test below asks you for both outcomes.

HELP WANTED

What we could not do, and how you can help

Four things need people or devices we do not have. We say what each one blocks and what we will do with what you send.

Windows, macOS, iOS and Android rendering. We have no Windows, Mac or phone in this study, and Android emulators need hardware virtualization our test machine lacks. Without them, this study cannot say what a Windows 10 PC, a Mac or a phone shows. The 30-second device test below is how a device you own becomes data. Send it from any phone or computer. If you can run the draw harness we publish on another system, or have a device lab, write to us.

Independent review of the detector. One reader (an AI model working through blind sheets) labelled the checks. A fresh sample of 274 cells and rating instructions are in the rater packet. It takes about an hour, needs no Unicode knowledge, and we publish agreement figures and the labels, credited to you if you want.

A real-device check of our missing-glyph test. The check was calibrated on Linux. The device test gives us the other outcome, the characters where it says *Shows* and you see a box, and the ones where it says *Missing* and you see a picture. We use both to recalibrate the check.

Android and Samsung emoji versions. We found no vendor page that names the Emoji version for any Android or Samsung release. If you have a device with a known system build, send the device test with the

build number.

Test your own device

This test draws 23 characters in your browser with the same check our site uses, and shows what the check concludes. Then you tell us what you actually see on screen: one picture means *Looks right*, an empty box, a box with digits, several small pictures in a row or a black outline where colour belongs means *Box or pieces*. A disagreement between the check and your eyes is the most useful result there is. Nothing is sent from this page. The buttons build a text for you to copy or email yourself, and it includes your browser name and system.

CHARACTER	WHAT YOU SHOULD SEE	WHAT OUR CHECK SAYS	WHAT YOU SEE
THE LETTER A (CONTROL) **ALWAYS**	A	**Shows**	○ Looks right ○ Box or pieces
GRINNING FACE (CONTROL) **EMOJI 1.0**	😀	**Shows**	○ Looks right ○ Box or pieces
RED HEART (CONTROL) **EMOJI 0.6**	❤️	**Shows**	○ Looks right ○ Box or pieces
VITHKUQI CAPITAL LETTER A **UNICODE 14.0**	𐕰	**Shows**	○ Looks right ○ Box or pieces
NAG MUNDARI LETTER O **UNICODE 15.0**	𑵃	**Shows**	○ Looks right ○ Box or pieces
TODHRI LETTER A **UNICODE 16.0**	𐗀	**Shows**	○ Looks right ○ Box or pieces
SIDETIC LETTER N01 **UNICODE 17.0**	𐑐	**Shows**	○ Looks right ○ Box or pieces
TOLONG SIKI LETTER I **UNICODE 17.0**	𑶀	**Shows**	○ Looks right ○ Box or pieces
BERIA ERFE CAPITAL LETTER ARKAB **UNICODE 17.0**	𖿠	**Shows**	○ Looks right ○ Box or pieces

CHARACTER	WHAT YOU SHOULD SEE	WHAT OUR CHECK SAYS	WHAT YOU SEE
TAI YO LETTER LOW KO UNICODE 17.0		Shows	○ Looks right ○ Box or pieces
SOM SIGN UNICODE 14.0		Shows	○ Looks right ○ Box or pieces
SAUDI RIYAL SIGN UNICODE 17.0		Shows	○ Looks right ○ Box or pieces
RUFIYAA SIGN UNICODE 18.0		Shows	○ Looks right ○ Box or pieces
UAE DIRHAM SIGN UNICODE 18.0		Shows	○ Looks right ○ Box or pieces
OMANI RIAL SIGN UNICODE 18.0		Shows	○ Looks right ○ Box or pieces
MELTING FACE EMOJI 14.0		Shows	○ Looks right ○ Box or pieces
SHAKING FACE EMOJI 15.0		Shows	○ Looks right ○ Box or pieces
PHOENIX (BIRD, ZERO-WIDTH JOINER, FIRE) EMOJI 15.1		Shows	○ Looks right ○ Box or pieces
FACE WITH BAGS UNDER EYES EMOJI 16.0		Shows	○ Looks right ○ Box or pieces
FIGHT CLOUD EMOJI 17.0		Shows	○ Looks right ○ Box or pieces
ORCA EMOJI 17.0		Shows	○ Looks right ○ Box or pieces
PEOPLE WRESTLING, TWO SKIN TONES EMOJI 17.0		Missing	○ Looks right ○ Box or pieces

CHARACTER	WHAT YOU SHOULD SEE	WHAT OUR CHECK SAYS	WHAT YOU SEE
CRACKING FACE **EMOJI 18.0**		**Shows**	◯ Looks right ◯ Box or pieces

Answer as many rows as you like. The controls (the letter A, the grinning face and the red heart) should all look right on any device. If one does not, tell us.

METHOD

Method

What was tested

Dataset 1: 59,852 code points of Unicode 8.0 to 18.0 from the Unicode Character Database, with release dates from Unicode's publication page. Dataset 2: 5,244 emoji sequence lines from the Unicode emoji data files, with Emoji release dates from UTS #51. Releases before 10.0 are controls, 10.0 to 18.0 the study range. The datasets are never pooled.

How an item was drawn

One item per fixed-size cell on a grid page, black on white at 48 px, in Chrome for Testing 155.0.8059.39 (headless, fixed flags, scale 1, language en-US), with the font set chosen by a fontconfig file. Firefox 155 and WebKit 26.6 (Playwright builds) were driven the same way. Firefox ran with its content-process sandbox off, because the sandbox hides fonts outside system folders.

How an item was classified

Six signals: an exact pixel match with the box the same browser draws for unassigned code points, the font Chromium reports through its developer protocol, that font's own character map (read with fontTools), the advance width, the glyph count and, for sequences, whether every part is covered by one font run. Classes are drawn as intended, box, hex-code placeholder, blank, monochrome where colour was meant, and for sequences composed, split into pieces, partial or missing. HarfBuzz (uharfbuzz) shapes sequences against font files with no browser.

Font history

45 Noto Color Emoji files from the font's public repository, 2017-03-10 to 2026-09-17, with commit date and the build date inside the file; monthly Noto family releases 2023-01-31 to 2026-09-30 for other scripts. Chromium 155 fixed, only the emoji font swapped.

Browser history

Chrome 113, 120, 128, 135, 141 and 155 against eight emoji files; 142 to 154 for the 60 sequences; 13 Chromium snapshot builds for the bisect. Fonts fixed.

Checking the detector

Two blind reviews of contact sheets with the predictions hidden, labelled by one reader (an AI model), not by independent people. Round 1 (362 items) was used to fix the detector, so its score is optimistic. Round 2 (274 items from a fresh seed, scored once with no revision): 189 single characters with 0 false positives

and 0 false negatives, and 85 emoji sequences with all 85 classes agreeing. The 95% upper bounds on the error rates are 4.0% and 3.8% for single characters. That is the honest figure, and independent raters are still wanted.

Re-measurement

We also redrew every code point and emoji sequence in a second harness that uses only the box-matching test, in the same Chromium build: it agrees with the full detector on 100% of left-to-right code points in two font sets and 99.98% in the third. For sequences, the simple width test agrees on 96.9%, 99.3% and 100% in the three sets, and the differences are black symbols a width test counts as composed. We then repeated the check on every item, not just the first 40 grid pages: 18 runs (device scale 2, a cold font cache, and a headed browser under a virtual display, over code points and sequences, in 3 font sets) compared 585,858 item readings with the main run. The supported-or-not decision changed in 0 of them.

Reproduction

We rebuilt both datasets from freshly downloaded Unicode files (byte-identical to the originals), rebuilt the container from the published recipe, and redrew Dataset 1 in it in two font sets. The class of every one of 60,600 items matched in the minimal set, and 60,599 of 60,600 in the Ubuntu packaged set. Pixels did not match (1% and 81% identical), because the host's font anti-aliasing defaults differ: DejaVu Sans's missing-glyph box drew 463 pixels of ink on the study machine and 446 in the container. The detector compares pixels only inside one run, so the classes are stable. Not rerun in the container: the newest-Noto set, the historical fonts and the Chrome matrix. The check also found a gap in our own recipe: the container has no package lists, so the script that fetches Ubuntu's Noto packages failed until we added an update step.

Statistics

Wilson intervals for shares. Releases, scripts and font files are the units that matter: items of one release share one font file, so item-level intervals are optimistic. Spearman correlations between release age and coverage have nine points and are indicative only.

Evidence tags

MEASURED from rendered pixels, font files or browser builds on the stated machine.
LINUX FONTS ONLY a property of this font set. **NOT TESTED** an open question.

LIMITS

Limitations

Ordered by how much each could change a conclusion.

One operating system image. Every rendering result is Chromium, Firefox or WebKit on Ubuntu 24.04 in a container, with a named font set. No statement is made about Windows, macOS, iOS, Android or about a share of devices. The Firefox and WebKit builds are the ones Playwright ships, not release builds of Firefox on Windows or Safari.

Rendering only. Keyboard and input availability, and text in browser menus and address bars, are different steps and were not measured.

One reader labelled the detector. The checks that validate the detector were labelled by one reader, an AI model, not by independent people. Independent raters are wanted (see Help wanted). Round 1 was used to tune the detector, so only round 2 is an honest estimate.

Pooled coverage mostly measures block size. Unicode 18.0 is 87% Seal, 10.0 is 88% and 13.0 is 83% Han extensions. Every headline also appears for a stratum without the huge blocks, and none of that removes the problem.

Age is release. There are nine distinct release ages, so “failure against age” is “failure against release”, and releases differ in content.

Items of one release are not independent. They share a font file and often a block, so item-level intervals are optimistic. For emoji the lag is 12 batch values, not 1,559 independent items.

Left and right censoring. The oldest emoji font is from 2017-03-10 and monthly Noto releases start in 2023. Anything supported then has no first-support date. Lags for recent releases are cut short because the window closes in 2026-10.

The commit date is not the build date. Fonts were sometimes committed months after they were built. Both are shown. A monthly tag is not the date a character first reached the font’s main branch.

A character map entry is not quality. A font that maps a code point may draw it badly. The monthly lag for scripts is at character-map level.

The Chromium bisect isolates ten commits. We did not rebuild Chromium with and without the segmentation change. The attribution rests on it being the only text-related commit in the window and on its message and test.

Marks and sequences. Combining marks were tested on a dotted circle in the main run and re-tested on real letters only for those that failed. The sequence test in other engines is a width test, so black symbols count as composed. Right-to-left characters are excluded from the cross-engine code point comparison.

Stacks are choices. With the same fonts, the CSS order changed results by up to 5 points. Our stacks are plausible, not exhaustive. OpenMoji was not tested, and only one version of each other font was.

Replays are not history. Swapping an old emoji font into a current browser isolates font evolution. It is not what a person saw in 2020 on a 2020 system.

Vendor dates are what vendors published. The vendor table comes from vendor release pages, not from drawing anything. Apple does not name the Emoji version, so those rows are inferred. We found no primary page for Android or Samsung, or for an Emoji 18.0 date for Apple or Microsoft.

We make a tool that depends on this. UltraTextGen publishes this study and offers the characters it tests. We found and report the defect in our own device check, and we want the real-device data.

What we do about each one: the device test and the rater packet turn the two largest gaps into data, every run is repeatable from the files below, and a later version keeps this address and says what changed.

RECORD

Corrections

Version 1.1, October 16, 2026. Three statements in version 1.0 did not match this study’s own data files. We found them when we recomputed the study from its published files. No conclusion changes.

- **Emoji font wait before Emoji 14.0.** Version 1.0 said the five releases before 14.0 had a median of 132 days. Six releases come before 14.0 (5.0, 11.0, 12.0, 12.1, 13.0 and 13.1), and their median is 106.5 days. The 132 was the median of five, with Emoji 13.1 left out. The 106.5 uses the later of Emoji 12.1's two font batches (273 days); with the earlier batch (18 days before the release) the median is 73.5. The 12.5 days for the six releases from 14.0 on is unchanged.
- **Chromium builds bisected.** Version 1.0 said 26 builds. The two bisects tested 13 builds (two-tone people wrestling) and 12 builds (emoji selector), 25 in all. Version 1.0 gave 13 for both.
- **Code points that fail in every condition.** Version 1.0 said 2,100 non-logographic code points. We cannot reproduce that figure from the class columns in the data files. The number not drawn as intended in any of the four font sets is 2,039. The earlier figure may have used a different definition (no Noto font maps the code point), which the published files do not let us check.

A clarification, with no change to any figure: the 55-day median is the median of 12 per-release values (54.5). The median across individual emoji sequences is 66 days.

Some statements rest on files we have not published (the character-map index of the installed fonts, and the Chrome 113 to 128 emoji-selector run on all 207 sequences), so we could not re-check them. They are unchanged and unverified.

DATA

Data, files and reproducibility

The files hold results for characters and fonts: counts, classes and dates. They hold no screenshots of vendor emoji, no personal data and nothing from the site's analytics. The rendering of every item is repeatable by anyone who runs the harness, and the per-item class files let you check any number on this page.

FILE	CONTENTS	SIZE
unicode-rendering-support-study.pdf	The full study, printed from this page	846 KB

Also published: the rater packet (instructions, a blank label sheet and eight contact sheets) and the harness, 54 scripts that build the datasets, draw every item, classify it and make every table. To rerun a layer, read REPRODUCE.md. The measurements were made on October 12 and 13, 2026. The source data (the Unicode files, Noto releases and Chrome builds) keeps changing, so a later rerun will differ and the harness records the version of each input.

CITE

How to cite this study

UltraTextGen Research. “How Fast Do New Unicode Characters and Emoji Show Up? A Browser and Font Rendering Study.” Version 1.1, October 16, 2026 (first published October 13, 2026). https://ultratextgen.com/research/unicode-rendering-support-study/

Please cite the version number. A later version will keep this address and say what changed.

Version 1.0 (October 13, 2026): first publication. It covers Chromium, Firefox and WebKit on Linux, 45 historical emoji fonts, 25 Chrome versions, the causes of two Chrome changes, other emoji font formats, a re-test of combining marks and the device check on our own site. Windows, macOS, iOS and Android are not measured; the device test and rater packet are how they will be added.

SOURCES

External sources

Unicode and Emoji release dates come from the Unicode Consortium’s publication dates page (https://www.unicode.org/history/publicationdates.html) and the release table in UTS #51, Unicode Emoji (https://www.unicode.org/reports/tr51/). The character and emoji data files are the consortium’s own, read from the latest published versions at the time of measurement.

When Windows 11 shipped each Emoji version is taken from Microsoft’s pages: KB5032190 (https://support.microsoft.com/en-us/topic/november-14-2023-kb5032190-os-builds-22621-2715-and-22631-2715-f9e3e13c-5e98-42c2-add8-f075841ca812)

for Emoji 15.0, KB5077241 (https://support.microsoft.com/en-us/topic/february-24-2026-kb5077241-os-builds-26200-7922-and-26100-7922-preview-b8cc7bc8-d640-4f18-9437-3ee59298b970)

for Emoji 16.0 and the Release Preview notes for build 26100.9539 (https://learn.microsoft.com/en-us/windows-insider/release-notes/release-preview-24h2-25h2/build-26100-9539-26200-9539) for Emoji 17.0. That Microsoft lists Emoji 15.1 only under Windows 11 comes from its Inside this update (https://support.microsoft.com/en-us/windows/inside-this-update-93c5c27c-f96e-43c2-a08e-5812d92f220d) page.

iPhone release dates come from Apple’s security releases page (https://support.apple.com/en-us/100100), and the emoji each release added from its release notes, for example iOS 26 (https://support.apple.com/en-us/123075). Apple does not state the Emoji version, so the pairing is ours.

The Chromium change is commit fb172d8c5b7d (https://chromium.googlesource.com/chromium/src/+/fb172d8c5b7d). Noto Color Emoji files and release history come from the noto-emoji repository

(https://github.com/googlefonts/noto-emoji), and the Twemoji colour font from Mozilla's twemoji-colr repository (https://github.com/mozilla/twemoji-colr).