Engineer

Die Unterstreichung durch jeden Buchstaben

28. September 2026

Unsere eigene Website unterstreicht jeden Link und lässt den Browser Lücken schneiden, wo eine Unterlänge die Linie kreuzt — bei `g`, `p` oder dem Komma. Safari tat das nicht. Die Linie lief gerade hindurch. Auf den englischen und dänischen Seiten geschah das schon so lange, wie die Website mehrsprachig war.

Die Ursache waren drei Schriftdeklarationen für ein Alphabet, das diese Seiten nie herunterladen.

Die technischen Erkenntnisse auf Ihr System anwenden

Die technischen Überlegungen hinter „Die Unterstreichung durch jeden Buchstaben“ können Ihre nächste Änderung unterstützen. Wir können ähnliches Verhalten in Ihrer Software, Ihren Daten oder Ihrer Infrastruktur untersuchen. Oberflächen, die schnell laden, auf dem Smartphone funktionieren und sich in allen Browsern Ihrer Kunden gleich verhalten.

Unsere Leistungen entdecken Kontakt aufnehmen →

Die Argumentation und ihre Grenzen nachvollziehen

Lesen Sie die Argumentation und ihre Grenzen in dieser Notiz. Wir können prüfen, ob dieselbe Diagnose in Ihrer Umgebung zutrifft. Full-Stack-GIS-Kartenentwicklung geleitet — Full-Stack-Kartenentwicklung von der Datenbank bis zum Browser

Full-Stack-GIS-Kartenentwicklung geleitet Kontakt aufnehmen →

Was der Browser tun sollte

`text-decoration-skip-ink` ist die Eigenschaft, die eine Unterstreichung um eine Unterlänge herum unterbricht. Sie ist standardmäßig aktiviert. Seit Jahren greift man bei unschönen Unterstreichungen zu `text-underline-offset`, um die Linie weiter nach unten zu schieben. Das ist der falsche Reflex: Das Verschieben verändert das Design; tatsächlich fehlte die Lücke.

Vor der interessanten Ursache fanden wir zwei andere echte Ursachen. Beide sollte man kennen.

- **Eine `1px` dicke Unterstreichung erhält eine `1px` große Lücke.** Die Unterbrechung der Rendering-Engine ist proportional zur Linienstärke. Unser Stylesheet legte ein Pixel fest; damit blieben auf beiden Seiten eines Buchstabenstamms ungefähr 0,6px Luft — technisch eine Lücke, optisch eine Linie durch den Buchstaben. `text-decoration-thickness: auto` skalierte beides gemeinsam und vergrößerte die Lücke von 3,5px auf 12,4px, ohne die Unterstreichung auch nur ein Pixel zu verschieben.
- `from-font` **ist nicht so sicher, wie es klingt.** Es liest die von der Schrift selbst angegebene Stärke. Ubuntu deklariert 0,056em für Light, 0,020em für Medium und 0,120em für Bold — eine Familie, deren Medium-Linie weniger als halb so stark ist wie die Light-Linie. Bei großen Schriftgraden stehen dadurch auf derselben Seite sechs Pixel neben zwei Pixeln Linienstärke.

Damit war Chrome korrigiert. Safari zog die Linie weiterhin durch alles hindurch.

Was am längsten dauerte

Wir maßen es gründlich: echtes Safari steuern, die Unterstreichung umfärben, einen Screenshot in achtfacher Skalierung aufnehmen und die Pixel an den Anfangs- und Endpunkten der Linie zählen.

	Gezeichnete Stärke	Lücke um eine Unterlänge
Chrome	2,00px	5,8 / 12,5 / 6,3px
Safari	1,38px	1,8 / 1,0 / 1,0px

Safari schnitt also eine Lücke. Sie war nur sechsmal schmaler als in Chrome, und bei normaler Lesegröße ist eine Ein-Pixel-Lücke keine erkennbare Lücke. Schlimmer noch: Der übliche Hebel verschärfte das Problem. Bei 3px Stärke fiel Safari von drei Lücken auf eine zurück, bei 4px auf keine, weil es die Lücke nicht mit der Linie skaliert. Sie bleibt in jeder Größe bei ungefähr 0,07em, während Chrome 0,6em liefert.

Wir prüften ungefähr zwanzig Deklarationen dagegen — `skip-ink: all`, beide alten Schreibweisen der Eigenschaft, alle Stärken von einem halben bis vier Pixeln, `text-underline-position`, `font-smoothing`, `text-rendering`, `paint-order`. Nichts verbreiterte die Lücke. Die ehrliche Schlussfolgerung lautete damals, WebKit begrenze sie einfach und ein Stylesheet könne nichts weiter ausrichten.

Diese Schlussfolgerung war falsch. Widerlegt wurde sie durch einen Fehlerbericht in einem Schriftprojekt: Safari ignoriert Skip-ink, wenn eine Schrift mit einem nichtlateinischen Zeichensatz geladen wird.

Die tatsächliche Ursache

Die Website liefert Ubuntu in sechs Teilen aus — drei Schriftstärken für Lateinisch und drei für Kyrillisch. So erscheint eine ukrainische Seite in derselben Schrift wie eine englische, statt auf eine Systemschrift zurückzufallen. Jeder Teil deklariert seinen Zeichenbereich; der Browser lädt nur die tatsächlich benötigten Teile herunter.

Alle sechs standen unter demselben Familiennamen, wie man es naheliegenderweise schreibt. Genau das ist der Auslöser, dokumentiert als WebKit-Fehler 255159: **Sobald ein Schnitt einer Familie einen nichtlateinischen Zeichenbereich deklariert, verschlechtert Safari Skip-ink für jedes Zeichen der Familie** — auch für lateinischen Text auf einer Seite, die die andere Datei nie abruft.

Eine englische Seite erhielt also schlechte Unterstreichungen, weil irgendwo im Stylesheet eine ukrainische Schrift deklariert war. Das Entfernen dieser drei Zeilen zur Laufzeit, ohne weitere Änderungen, vergrößerte die Lücken von 1,8/1,0/1,0px auf 4,5/11,0/5,0px.

Die Lösung

Geben Sie dem zweiten Schriftsystem einen eigenen Familiennamen und nennen Sie es im Schriftstapel hinter dem ersten:

```
@font-face {
    font-family: UbuntuCyrillic;   /* was: Ubuntu */
    src: url(ubuntu_300_cyrillic.woff2) format('woff2');
    unicode-range: U+0400-045F, U+0490-0491;
}

:root {
    --body-font: Ubuntu, UbuntuCyrillic, system-ui, sans-serif;
}
```

Ein lateinischer Buchstabe wird in der ersten Familie gefunden und erreicht die zweite nie. Ein kyrillischer fällt an der ersten vorbei, die keine entsprechende Glyphe enthält, und landet in der zweiten statt in einer Systemschrift. Die Zeichenbereiche entscheiden weiterhin über die Downloads; am Ladeverhalten ändert sich nichts. Englische und dänische Seiten laden drei lateinische Dateien und keine kyrillische, ukrainische Seiten umgekehrt.

Drei umbenannte Deklarationen und ein Eintrag im Schriftstapel. Safaris Lücken wuchsen auf 4,5/11,0/5,0px. Chrome blieb unverändert, die Unterstreichung bewegte sich nicht, und ukrainischer Text wird weiterhin in Ubuntu mit identischen Breiten gesetzt.



Was wir der nächsten betroffenen Person sagen würden

- **Eine Skip-ink-Messung gilt nur für die Engine, die sie erzeugt hat.** Wir hatten eine Zahlentabelle, eine korrekte Diagnose und eine echte Lösung — alles für Chromium. Der Screenshot, der den Fall wieder öffnete, kam von einem Menschen, der die Seite ansah.
- **Zuerst die Stärke prüfen, dann den Abstand.** Eine breitere Lücke erhält das Design; eine nach unten verschobene Linie ersetzt es.
- **Die Aufteilung nach Schriftsystem ist richtig, muss aber auch im Familiennamen stehen.** Ein gemeinsamer Name liest sich besser, kostet in Safari aber Skip-ink. Die Zusammenführung sieht nach Aufräumen aus und ist eine Regression.
- **Das Laden war nie das Problem.** Der erste Verdacht bei einer durch kyrillische Schriften gestörten Seite ist ein unerwünschter Download. Den gab es nicht. Die Deklaration allein genügte.

Quellen

- WebKit-Fehler 255159 — `text-decoration-skip-ink` und Schriftteilsätze
- Fontsource-Issue 1096, das den Auslöser benannte
- MDN: `text-decoration-skip-ink`
- MDN: `unicode-range`

Alle Blogbeiträge

Messen, bevor Sie ändern

Beschreiben Sie den Arbeitsablauf, die vorhandenen Schnittstellen und die Nutzer, die die Veränderung benötigen. Wir können eine prüfbare Lieferung definieren und die Übergabe so gestalten, dass Ihr Team die gelieferten Bestandteile warten kann.

Preise ansehen Kontakt aufnehmen →

Ihre Beobachtungen besprechen

Besprechen Sie „Die Unterstreichung durch jeden Buchstaben“ im Kontext Ihres eigenen Projekts. Erzählen Sie uns, was Sie beobachtet haben und wie ein nützliches Ergebnis aussehen würde.

Unsere Leistungen entdecken

welcome@engineer.company +45 50 10 61 84 Kopenhagen, Dänemark

Nehmen Sie diese Seite mit

Kontaktkarte

Mit Sorgfalt und Liebe gestaltet von D.

CVR 45969703 · Tagensvej 105C, 3. th, 2200 København N

Copyright © 2025 – heute · Engineer ApS · Alle Rechte vorbehalten.

Gedruckt von https://engineer.company/de/notes/safari-underline-font-subset/