

# Bundesnetzagentur will digitale Plattformen beaufsichtigen



## Markt + Trends | World Wide Web

Unternehmen bezahlen Pentester dafür, dass sie versuchen, in ihre Systeme einzubrechen. So dubios das für Außenstehende klingen mag – Penetrationstests sind seit Jahren ein etabliertes Mittel, die Sicherheit von Systemen und Netzwerken zu überprüfen. Im Folgenden erfahren Sie, wie ein sogenannter Bl...

Die Bundesnetzagentur (BNetzA) bringt sich für die Rolle des nationalen Koordinators für Digitalisierung nach dem Digital

Services Act (DSA) in Stellung. Dieser soll Onlineplattformen mit mehr als 45 Millionen aktiven Nutzern beaufsichtigen und als Schnittstelle zur EU-Kommission dienen. In einer Onlinediskussion bekräftigte BNetzA-Vizepräsident Wilhelm Eschweiler, seine Behörde sei für die Plattformaufsicht gut gerüstet, man habe Informatiker und Datenwissenschaftler eingestellt und sei auf dem Weg, sich von einer sektorspezifischen Aufsichtsbehörde zu einer Digitalagentur zu entwickeln.

Auch sei die BNetzA bereits jetzt dafür zuständig, Bußgelder gegen Provider zu verhängen, und sie sei politisch unabhängig. Derzeit sind viele konkrete Zuständigkeiten bei der Umsetzung des DSA noch offen. Eschweiler sprach von einem „Schaulaufen der Behörden“, das derzeit stattfindet. ([ulw@ix.de](mailto:ulw@ix.de))

---

## **Chrome-Browser bekommt Root-CA-Speicher**



## Markt + Trends | World Wide Web

Unternehmen bezahlen Pentester dafür, dass sie versuchen, in ihre Systeme einzubrechen. So dubios das für Außenstehende klingen mag – Penetrationstests sind seit Jahren ein etabliertes Mittel, die Sicherheit von Systemen und Netzwerken zu überprüfen. Im Folgenden erfahren Sie, wie ein sogenannter Bl...

Google hat seine Pläne für eine im Chrome-Browser eingebaute Verwaltung von Root-CAs konkretisiert und das bereits 2020 angekündigte Chrome Root Program offiziell gestartet. Bislang verwendet Chrome für die Überprüfung von HTTPS-Zertifikaten das Betriebssystem als Vertrauensanker und lässt alle Zertifikate gelten, deren digitale Unterschrift sich auf eines der vom Betriebssystem installierten Root-CA-Zertifikate zurückführen lässt. Künftig soll ein Chrome Root Store im Browser diese Zertifikate verwalten. Google bestimmt dann selbst, welche Root CA (Certification Authority) als

vertrauenswürdig gilt, und verstärkt so seinen Einfluss auf die Herausgeber der Root CAs.

Chrome Root Store soll lokal verwaltete Zertifikate berücksichtigen. Wenn also ein Zertifikat zum Beispiel via Windows Group Policy verteilt wird, soll es Chrome als vertrauenswürdig einstufen. Den Übergang will Google allmählich vollziehen, bereits der im September erschienene Chrome 105 wird laut Google für manche Nutzer mit aktiviertem Root Store ausgeliefert. Wer das Feature testen will, kann beim Browserstart ein entsprechendes Flag setzen. ([ulw@ix.de](mailto:ulw@ix.de))

---

## **Ethereum-Blockchain arbeitet jetzt mit Proof of Stake**



## Markt + Trends | World Wide Web

Unternehmen bezahlen Pentester dafür, dass sie versuchen, in ihre Systeme einzubrechen. So dubios das für Außenstehende klingen mag – Penetrationstests sind seit Jahren ein etabliertes Mittel, die Sicherheit von Systemen und Netzwerken zu überprüfen. Im Folgenden erfahren Sie, wie ein sogenannter Bl...

Die Ethereum-Blockchain hat den Wechsel zum Proof-of-Stake-Verfahren offiziell vollzogen. Bei dem als Merge bezeichneten Übergang integrierten die Ethereum-Entwickler die Beacon Chain, auf der man das Konsensverfahren seit 2020 erprobte, mit dem Ethereum Mainnet. Der erste Schritt des lange erwarteten Merge war ein Update der Beacon Chain Anfang September. Nach Erreichen der vorher bestimmten Gesamtschwierigkeit für das Erzeugen neuer Blöcke fand das Mining von Ethereum Mitte September ein Ende.

Statt mit Proof of Work am schnellsten einen mathematischen Beweis auszurechnen, erzeugt nun ein ausgeloster Validierer den neuen Block. Um einen Block zu validieren (zu staken), muss man einen Full Node betreiben und eine Sicherheitssumme in der Kryptowährung von Ethereum hinterlegen. Derzeit beträgt die Sicherheit mindestens 32 Ether, die dem Besitzer im Falle eines Betrugs aberkannt werden. Durch den Wechsel des Konsensverfahrens entfällt das umweltschädliche Mining. Der Energiebedarf der Ethereum-Blockchain soll sich damit um 99,95 Prozent verringern. Im Zuge des Merge fiel der Preis von Ether um mehr als 20 Prozent auf etwa 1330 US-Dollar. ([pst@ix.de](mailto:pst@ix.de))

---

## **Cloudflare will mit Turnstile reCAPTCHA ablösen**

**Mit dem neuen Human-Validierungsdienst Turnstile will Cloudflare bisher gängige CAPTCHA-Verfahren ersetzen. Das soll die User Experience verbessern.**



## Markt + Trends | World Wide Web

Unternehmen bezahlen Pentester dafür, dass sie versuchen, in ihre Systeme einzubrechen. So dubios das für Außenstehende klingen mag – Penetrationstests sind seit Jahren ein etabliertes Mittel, die Sicherheit von Systemen und Netzwerken zu überprüfen. Im Folgenden erfahren Sie, wie ein sogenannter Bl...

Cloudflares CAPTCHA-Alternative Turnstile soll das Suchen nach Zebrastreifen, Bussen und Ampeln für Nutzer in Zukunft obsolet machen. Turnstile ist in der offenen Beta und soll für Internet-User unsichtbar deren Menschsein bestätigen. Dafür soll Turnstile die Validierung automatisch aus einer Reihe von nicht intrusiven Browseraufgaben vollziehen – basierend auf der Telemetrie und dem Kundenverhalten während einer Sitzung.

Ein weitere Validierungsmöglichkeit bei Turnstile sind Private Access Tokens, die die Gerätehersteller vergeben und mit denen

sich das Gerät als legitim ausweist. Das Token bürgt dabei für die Validität der Nutzer. Das will Cloudflare über Kollaborationen mit Herstellern erreichen, Apple sei bereits an Board. Damit könne man die Datenerhebung für Turnstile minimieren: Statt Geräte selbst auszulesen, erledige das nun der Hersteller.

## **Bestehende Dienste bei Nutzern unbeliebt**

Neben der Konkurrenz zum eigenen Dienst CAPTCHA, bekannt vom Klick auf „Ich bin ein Mensch“, hofft Cloudflare Googles reCAPTCHA ablösen zu können. In der Bekanntmachung der Beta von Turnstile hebt Cloudflare vor allem die schlechte User Experience der bestehenden Validierungsdienste hervor: „Wir hassen es, ihr hasst es, alle hassen es.“

Dabei ist Turnstile nicht der erste Validierungsdienst, der keine Interaktion erfordert. Auch Googles reCAPTCHA existiert bereits in einer Variante, die im Verborgenen basierend auf dem Benutzerverhalten eine Entscheidung trifft. Zum Einsatz kommt diese reCAPTCHA-Version aber selten. Cloudflare hebt ihr gegenüber die komfortablere Bedienung des eigenen Produkts hervor.

Turnstile steht ab sofort allen Webseitenbetreibern kostenlos über eine API zur Verfügung. Cloudflare-Kunde muss man nicht sein. Die Anmeldung erfolgt über eine eigene Website. ([jvo@ix.de](mailto:jvo@ix.de))

---

## **Eine Rechtsverordnung soll**

# Cookie-Abfragen eindämmen

## Cookie-Banner adieu?

# Eine Rechtsverordnung soll Cookie-Abfragen eindämmen

Die Bundesregierung will zentral verwaltete Cookie-Einstellungen ermöglichen. Nutzer könnten sogar pauschal alle Cookies ablehnen. Allerdings würden sie dann wohl mit Hinweisbannern zu kostenpflichtigen Angeboten überschwemmt.

Von Holger Bleich

Die Datenschutz-Grundverordnung (DSGVO) lässt derzeit keinen Ausweg: Website-Betreiber müssen ihre Besucher um Erlaubnis fragen, bevor sie Tracking- oder Analyse-Cookies auf deren Rechner setzen. Diese Einwilligung muss informiert erfolgen, weshalb nahezu jede werbefinanzierte Website Pop-up-Banner vorschaltet, die viel Text enthalten. Und viele Banner stupsen Nutzer mit Design-Tricks zum „Ja“.

Das im Dezember 2021 in Kraft getretene Telekommunikation-Telemedien-Datenschutz-Gesetz (TTDSG) sieht in seinem Paragraphen 26 einen deutschen Sonderweg vor, der die nervigen Banner überflüssig machen soll: In „anerkannten Diensten zur Einwilligungsverwaltung“ hinterlegen demnach Nutzer ihre Cookie-Präferenzen zentral. Websites fragen dann nicht mehr direkt den User, sondern holen sich Einwilligung oder Ablehnung bei dem Dienst ab [1].

Nun hat das von Volker Wissing (FDP) geführte Bundesministerium für Digitales und Verkehr (BMDV) den Entwurf zu einer Verordnung erarbeitet, die den nach TTDSG

erforderlichen Rechtsrahmen für die Einwilligungsdienste definieren soll. Nutzer können dem Entwurfstext zufolge „generelle Einwilligungen geordnet nach Kategorien für bestimmte Zugriffe auf Endeinrichtungen und Gruppen von Telemedienanbietern erteilen“. Die Dienste müssen gut erklären und informieren, außerdem sollen sie den Nutzer nicht mit Voreinstellungen beeinflussen.

Im BMDV glaubt man, im Entwurf die „richtige Balance“ zwischen den Interessen von Nutzern und kommerziellen Anbietern getroffen zu haben. Es gebe „keinen Anspruch auf kostenlosen Content“, war aus dem Ministerium zu hören. Diese Prämisse spiegelt sich im Entwurf wider: Zwar dürfen Nutzer im externen Einwilligungsdienst das Setzen von Cookies generell ablehnen. Doch gestattet es der Verordnungsentwurf Anbietern in diesem Fall, Nutzer mit vorgeschalteten Bannern darauf hinzuweisen, dass sie Tracking-Cookies benötigen, um die Site über Werbung zu finanzieren. Außerdem dürfen sie auf ein „kostenpflichtiges Alternativangebot“ (auf Medien-Websites die sogenannten „Pay-Per-View“) verweisen oder den Nutzer „zur Änderung seiner Voreinstellungen beim Dienst zur Einwilligungsverwaltung“ auffordern.

Wie Websites die Nutzerpräferenzen beim Einwilligungsdienst abfragen sollen, lässt der Entwurf offen. In der Begründung zum Text, die c't vorliegt, spricht das BMDV von „technikneutral“. Der Browser könne etwa einen HTTP-Request schicken, der die Zusatzinformation enthalte, dass der Endnutzer einen Dienst zur Einwilligungsverwaltung verwendet.

Auch wie die Dienste selbst funktionieren, will das BMDV dem Markt überlassen. Sie dürfen „kein wirtschaftliches Eigeninteresse“ daran haben, dass Nutzer möglichst viele Einwilligungen erteilen. Wohl aber dürfen sie kommerziell agieren und auch Geld für ihre Services verlangen. Sie müssen ein Sicherheitskonzept vorweisen und sich anschließend von der Bundesdatenschutzbehörde prüfen und zertifizieren lassen.

Den ersten Entwurf hat das BMDV Ende August mit Bitte um Stellungnahmen an Wirtschaftsverbände geschickt. Er ist noch nicht mit anderen Bundesministerien abgestimmt. Bis er im Bundestag landet, werden noch viele Änderungen folgen – und Monate vergehen. Sollte der Bundestag die Rechtsverordnung durchwinken, muss die EU im sogenannten Notifizierungsverfahren prüfen, ob der Text europarechtskonform ist.

## **„Handwerkliche Fehler“**

Daran regen sich bereits Zweifel. In einer ersten Stellungnahme kritisiert der Bundesverband Digitale Wirtschaft (BVDW) „handwerkliche Fehler“ im Text und moniert, dass der „aktuelle europäische Rechtsrahmen nicht hinreichend gewürdigt“ werde. Der Gesetzgeber habe „in den letzten Jahren zunehmend darauf hingewirkt“, die Einwilligung als Rechtsgrundlage für Datenverarbeitung zu forcieren, und nun wolle er „deren Einholung quasi untersagen“. Dies sei „aus Sicht der Datenökonomie und besonders aus Sicht der informationellen Selbstbestimmung der Nutzerinnen und Nutzer nicht der große Wurf, sondern ein großer Rückschritt“.

Das BMDV hat seinen Entwurf genau in einer Zeit vorgelegt, in der die EU dabei ist, ohnehin Cookie-Vorgaben und Einwilligungserfordernisse neu zu regeln: In Brüssel arbeiten Rat, EU-Parlament und Kommission gerade an einem Kompromiss zur E-Privacy-Verordnung. Weil diese EU-Verordnung über dem deutschen Recht stehen wird, könnte die deutsche Rechtsverordnung je nach Verhandlungsergebnis also schon in zwei bis drei Jahren wieder obsolet sein. ([hob@ct.de](mailto:hob@ct.de))

1. Literatur
2. [Holger Bleich, Löcher stopfen per Verordnung, Die bizarre Tracking-Regulierung in Deutschland, c't 16/2022, S. 34](#)

---

# Suchmaschinen

## Neue Wegweiser

# Kagi, Neeva, You.com: Die neuen Google-Konkurrenten ausprobiert

Wer im Internet sucht, der googelt. Mehrere Suchmaschinen-Start-ups wollen das ändern. Ihre Suchdienste sind frei von Werbung und Trackern und bieten mehr Personalisierung, Datenschutz, Spezialsuchen sowie viele weitere nützliche Funktionen.

Von Jo Bager

Seit vielen Jahren dominiert Google den Suchmaschinenmarkt mit einem weltweiten Anteil von rund 90 Prozent. Der nächstkleinere Konkurrent Bing, immerhin betrieben von Microsoft, kommt je nach Marktforschungsunternehmen auf nur drei bis fünf Prozent. Die Dominanz Googles hat drei junge, allesamt US-amerikanische Unternehmen aber nicht abgeschreckt, neue Suchdienste an den Start zu bringen: Kagi, Neeva und You.com.

## Neeva

[Neeva](#) wurde von zwei ehemaligen Google-Mitarbeitern aus der Taufe gehoben, die damit unzufrieden waren, wie Google seine Suchergebnisseiten mit Werbung überfrachtet. Neeva ist daher

werbefrei und soll sich stattdessen über ein Freemium-Modell finanzieren. Ohne Third-Party-Cookies kommt die Desktopversion allerdings nicht aus. Neevas Bedienoberfläche ist auch auf deutsch verfügbar.

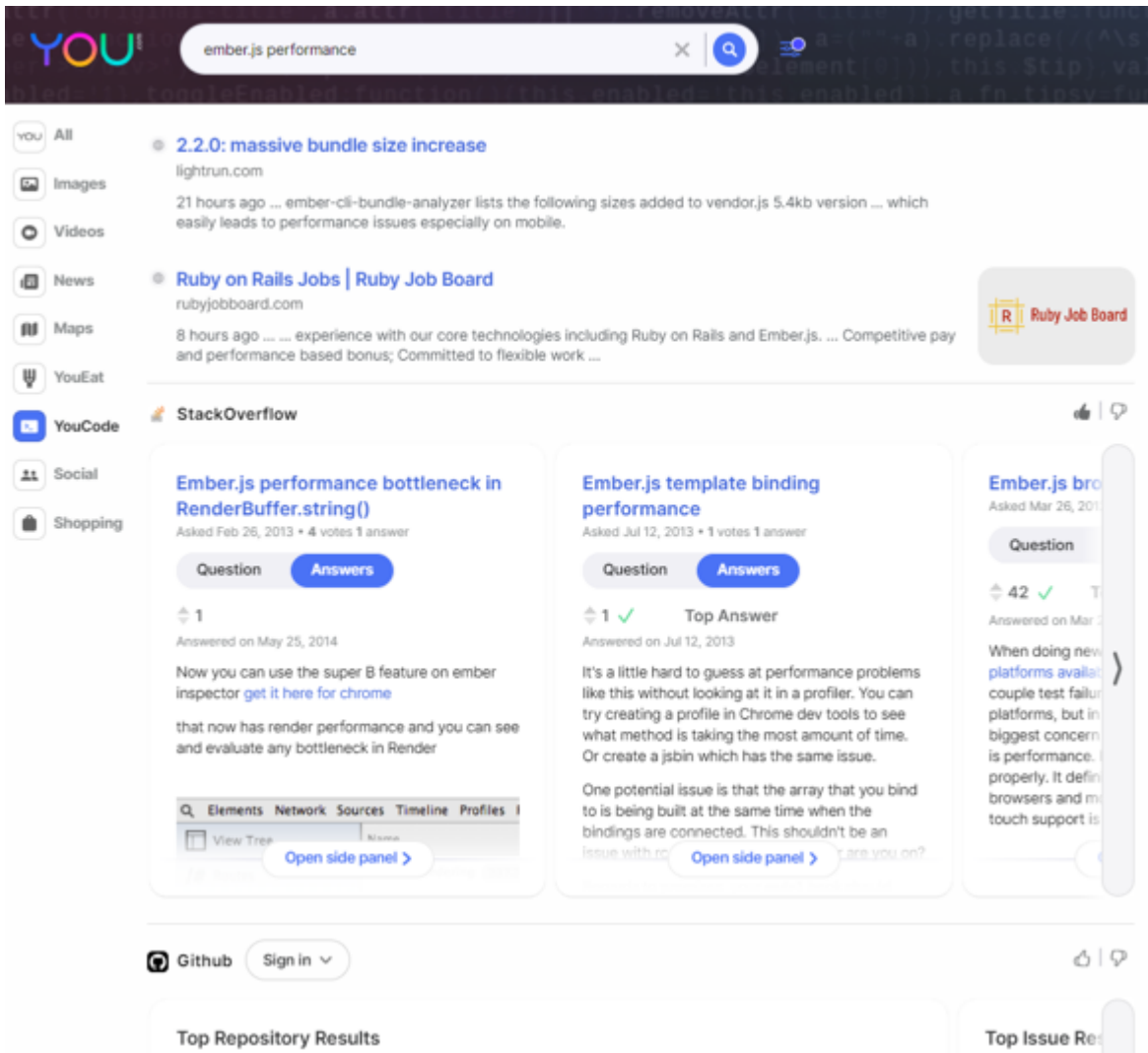
Die Betreiber bauen einen eigenen Suchmaschinenindex auf, nutzen derzeit aber auch Ergebnisse von Bing. Bei bestimmten Themen bietet Neeva von sich aus themenspezifische Filter an, um die Suche zu verfeinern. Wer zum Beispiel nach einem Programmierthema sucht, der kann die Ergebnisse auf „Official Docs“, „Forums“, „Programming Websites“, „Blogs“ und „Code Repos“ einschränken.

Man kann Neeva wie andere Suchmaschinen anonym nutzen. Viele der Besonderheiten erschließen sich allerdings nur Nutzern mit Account. Ein Basis-Account ist kostenlos, Premium-Accounts für 5 US-Dollar im Monat enthalten Zugänge für das BitDefender-VPN und den Passwortverwalter LastPass. Die Premiumvariante ist derzeit allerdings nur in den USA verfügbar.

Eingeloggten Nutzern stellt Neeva ein persönliches Dashboard zusammen, mit Informationspanels zum Beispiel zu Aktienkursen und dem lokalen Wetter sowie einem Newsfeed mit einer individualisierbaren Auswahl an Quellen. Hinter der Login-Schranke können Nutzer ihre Treffer-Links in sogenannten Bereichen speichern. Und sie können festlegen, von welchen Quellen sie mehr oder weniger Treffer erhalten möchten. Vor allem aber können sie ihre bei bestimmten Clouddiensten gespeicherten Daten durchsuchen lassen. Dazu verknüpfen sie ihre Google-, Microsoft- oder Dropbox-Konten mit Neeva – sehr praktisch.

Neeva bietet einen Browser mit integrierter Suchmaschine und Tracker-Blocker als App für Android und iOS an. Schon während der Benutzer dort einen Suchbegriff eintippt, macht ihm die App Vorschläge für Ziel-Sites. Die iOS-App bietet zudem eine weitere Funktion, NeevaScope, die weiterführende Informationen zur aktuell besuchten Site präsentiert, etwa ähnliche Sites.

Der Betreiber hat auch Neeva-Erweiterungen mit Tracker-Blockern für Firefox, Safari und Chromium-Browser veröffentlicht. Unter Neeva.xyz betreibt er eine Suchmaschine für das Web3.



You.com zeigt die Treffer ausgewählter Quellen prominent an.

## You.com

[You.com](https://www.you.com) sticht durch seine ungewöhnliche Oberfläche aus dem Suchmaschinen-Einerlei hervor. Statt oberhalb der Suchergebnisse ordnet der Dienst Reiter für Suchen nach spezifischen Medientypen und Themen links an. Bei den News- und Video-Suchergebnissen füllt You.com den gesamten zur Verfügung stehenden Platz mit Ergebniskacheln.

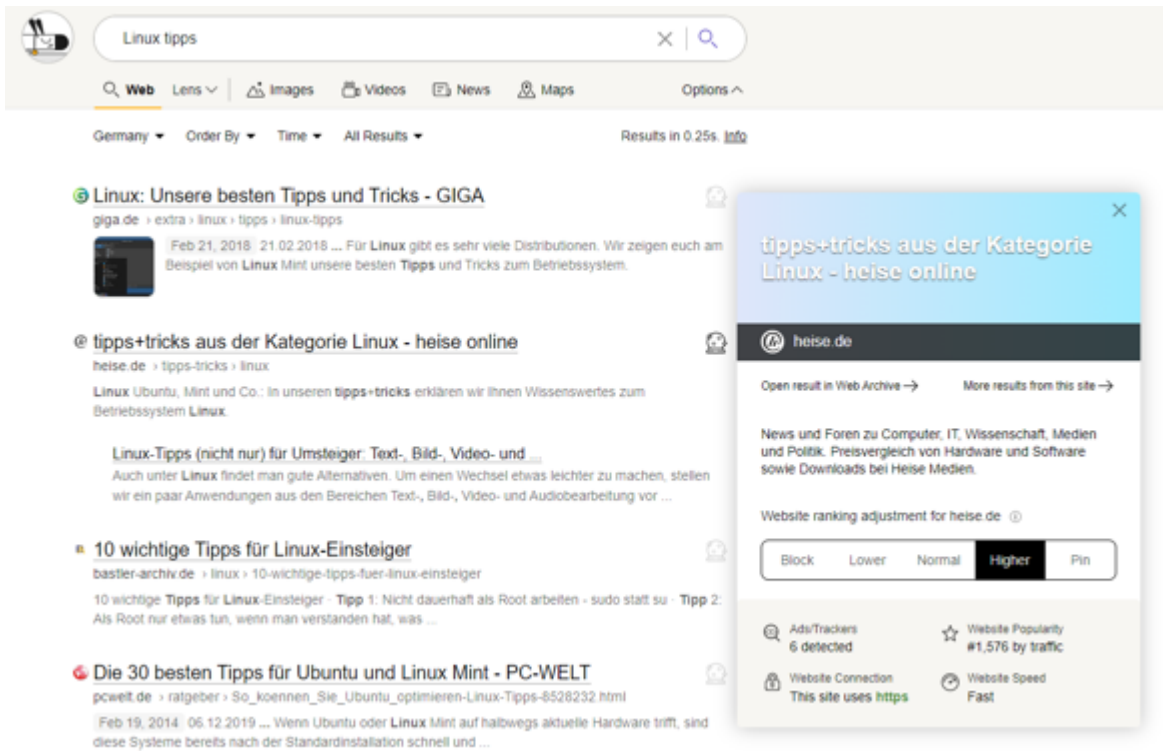
Die Haupt-Suchergebnisseite ist sehr aufgeräumt gestaltet. Je nach Abfrage streut You.com eine oder mehrere Listen mit

Kacheln ein, die Inhalte aus bestimmten Quellen enthalten (Bühnen). Solche Quellen nennt You.com Apps. Eingeloggte Benutzer können für mehr als 150 solcher Websites festlegen, ob sie mehr oder weniger Inhalte zu sehen bekommen wollen.

Vorwiegend aus den App-Inhalten speisen sich einige Spezialsuchen von You.com, zum Beispiel „YouCode“ für die Recherche auf Entwicklerplattformen, „Social“ und „Shopping“. An „YouEat“ für die Suche nach US-amerikanischen Bringdiensten und an der englischsprachigen Oberfläche zeigt sich, dass sich You.com derzeit vor allem an ein US-amerikanisches Publikum wendet. Mit einem prominent auf der Startseite verlinkten Goodie, YouWrite, kann man eine KI kurze englischsprachige Texte verfassen lassen. Für deutsche Nutzer gibt es immerhin schon neun deutsche News-Quellen, von Bild.de über Tagesschau bis T-Online.de.

You.com ist auch ohne Account – und ohne individuelle Anpassungen – nutzbar. Ob mit oder ohne Account: Der Dienst ist derzeit kostenlos und werbefrei. Allerdings hat der Gründer Richard Socher in einem Interview die Möglichkeit erwähnt, You.com durch sogenannte Private Ads zu refinanzieren – Werbung, die dem Werbenden nichts über den Surfer verrät und auch kein Tracking zulässt.

Einen Teil der Suchergebnisse bezieht You.com von Bing. Für themenspezifische Abfragen verfügt die Suchmaschine über eigene Indizes. Der Betreiber stellt Apps für Android und iOS sowie Add-ons für Firefox und Chromium-Browser bereit.



Mehr hiervon, bitte: Bei Kagi gewichtet man Websites von Hand und beeinflusst so das Ranking.

## Kagi

Um bei [Kagi](#) zu suchen, muss man einen Account einrichten, denn der Dienst finanziert sich mit einem Freemium-Preismodell. Kostenlos sind 50 Suchabfragen pro Monat; wer Kagi unbeschränkt nutzen will, zahlt monatlich 10 US-Dollar. Kagi ist komplett werbe- und trackerfrei.

Kagi bietet wie kein anderer Dienst die Möglichkeit, das aufgeräumte Erscheinungsbild anzupassen, vom Farbschema über die Schriftgröße bis zur Anzeige der Favicons der gefundenen Websites. Per Default sind Inline-Bilder, -News und -Videos, Sofortantworten, verwandte Suchen und viele weitere Elemente aktiviert, die man von anderen Suchmaschinen kennt. Man kann sie bei Kagi aber deaktivieren.

Kagi lässt den Nutzer zudem händisch in das Ranking eingreifen. Neben jedem Suchergebnis zeigt der Dienst dazu eine Kristallkugel an. Klickt der Nutzer darauf, kann er für die Zukunft Inhalte der betreffenden Domain blockieren, herab-

oder heraufstufen.

Suchergebnisse lassen sich mit sogenannten Lenses filtern. Sie schränken die Resultate auf bestimmte Dateitypen, Regionen, Zeiträume und Sites ein oder schließen bestimmte Websites oder Keywords aus. Solche Lenses lassen sich auch selbst einrichten – benutzerdefinierte Suchmaschinen. Wer vom Kagi-Suchformular aus gezielt bei anderen Diensten suchen will, kann dafür Kurzbefehle nutzen, sogenannte Bangs (die ursprünglich von DuckDuckGo stammen).

Kagi unterhält einen eigenen Index für Webseiten und News. Die Suchmaschine bezieht Ergebnisse aber auch (anonymisiert) von Konkurrenten wie Google und Bing und vertikalen Quellen wie Wikipedia ein. Der Betreiber stellt Add-ons für Firefox, Safari und Chromium-Browser bereit.

## Fazit

Wie wohltuend aufgeräumt eine Suchergebnisseite ohne Werbung ist! Und es ist auch sehr beruhigend, wenn der Suchdienst die Recherchen im Internet nicht trackt. Eine weitere wichtige Gemeinsamkeit der drei Anbieter ist die Anpassbarkeit der Suchergebnisse: Man ist nicht mehr auf Gedeih und Verderb einem Algorithmus ausgeliefert, sondern kann ein Stück weit das Ranking mitbestimmen.

Alle drei Dienste bereichern den Suchmaschinen Sektor noch um weitere gute Ideen. Es macht Spaß, sie auszuprobieren. Bleibt zu hoffen, dass die jungen Unternehmen tragfähige Geschäftsmodelle finden – vielleicht gibt es ja genug erfahrene Nutzer, die nicht mehr einfach nur die Ergebnisse hinnehmen wollen, die Google ihnen liefert. ([jo@ct.de](mailto:jo@ct.de))

**Links zu den Diensten:** [ct.de/y9nx](https://ct.de/y9nx)

---

# Handytickets für iOS erzeugen

## Taschenticketautomat

# Apple Wallet: Handytickets für iOS erzeugen

Fluggesellschaften bieten sie an, die Bahn und Konzertveranstalter: digitale Tickets, die man nicht ausdrucken muss und einfach auf dem Mobiltelefon mitnimmt. Apple hat sich dafür die App namens Wallet und ein Dateiformat mit raffinierten Zusatzfunktionen ausgedacht. So erzeugen Entwickler mit überschaubarem Aufwand eigene digitale Tickets, Gutscheine und Kundenkarten.

Von Jan Mahn

### **kompakt**

- Digitale Tickets auf dem Handy sind praktisch und mehr als ein Ersatz für die Papierversion. Sie sind schnell griffbereit und informieren zum Beispiel über Planänderungen.
- Apple hat für iOS eine Ticket-App namens Wallet zusammen mit einem eigenen Datenformat entwickelt.
- Um als Entwickler Tickets zu erzeugen, braucht man theoretisch nur einen Texteditor und einen Apple-Entwickleraccount – für echte Tickets vom Fließband gibt es Open-Source-Generatoren.

Früher, eigentlich bis vor wenigen Jahren, waren sämtliche

Eintrittskarten und Reisetickets noch durchgängig vom Anbieter gedruckte Dokumente, auf einem speziellen Papier mit Glitzerstreifen, damit sie nicht jeder nachbauen konnte. Als nächste Evolutionsstufe folgten Tickets zum Selbstaussdrucken. Die Echtheit bestätigt man seither nicht mehr über ein exklusives Spezialpapier, sondern über einen aufgedruckten Code, sei es ein Bar- oder ein QR-Code. Das Kontrollpersonal scannt ihn und ein mit einem Server verbundenes Lesegerät verrät, ob dieses Ticket gültig und noch nicht genutzt ist.



Anstatt den Code auszudrucken und ein zerknicktes A4-Ticket zum Konzert mitzunehmen, kann man den Code inzwischen auch direkt auf dem Mobiltelefon belassen und dessen Display beim Einscannen vorzeigen. Doch wer schon mal versucht hat, ein PDF-Ticket auf dem Handy aus der Mail-App zu fischen und so zu zoomen, dass man den QR-Code bequem scannen kann, der weiß, dass das nicht an den Komfort von Papier herankommt. Bequemer geht es mit einer Ticket-App. Apple hat sich schon 2012 die App Passbook ausgedacht und später in Wallet umbenannt. Neben Tickets liegen dort seitdem auch Apple-Pay-Kreditkarten. Die App funktioniert aus Kundensicht denkbar einfach: Man bekommt auf welchem Weg auch immer – per Mail, in einer anderen App oder auf einer Website – einen Link, um ein Ticket herunterzuladen oder direkt eine Ticket-Datei. Auf dem iOS-Gerät wird die Datei direkt in der Wallet-App geöffnet und die Karte dem Wallet hinzugefügt. Öffnet man den Link auf einem per iCloud verbundenen macOS-Computer, landet sie ebenfalls via Cloud-Magie auf dem iOS-Mobiltelefon. Und auch auf die Apple-Watch rutschen die digitalen Karten auf Wunsch. Gegen Vervielfältigung und Weitergeben sind die Pässe nicht geschützt, es können auch mehrere Konzertbesucher dasselbe Ticket auf ihr Telefon laden – so wie man auch ein PDF-Ticket mehrfach ausdrucken kann. Beim Einlass muss der Bar- oder QR-Code also immer mit einer Datenbank abgeglichen werden.



SITZPLATZ  
Bierbank 1

VERANSTALTUNG  
Grillfeier mit Freunden

ORT  
Garten

DATUM  
13.8.2022

MITBRINGEN  
Nudelsalat

UNVERTRÄGLICHKEITEN  
Nüsse

HUNGER  
mittel



Die Einladung zu einer fiktiven Grillfeier mit Freunden steckt als digitales Ticket in der Apple-Wallet-App: Für Freunde mit Android-Telefonen muss man entweder ein eigenes Ticket ausstellen oder ihnen eine alternative App empfehlen.

Beim Einlass zum Konzert findet der Besucher all seine gültigen Tickets fein säuberlich sortiert in dieser App und der QR- oder Barcode ist immer an den Bildschirm angepasst. Doch damit nicht genug: Der Ticketanbieter kann das Ticket auch so präparieren, dass es zum Beispiel eine Stunde vorm Konzert beim Benutzer aufpoppt und er es nicht einmal mehr raussuchen muss. Und wenn die Show ausfällt oder sich der Flug verspätet, kann der Veranstalter alle Kunden mit Wallet-Ticket per Push-Benachrichtigung informieren, ohne dass diese eine separate App bräuchten. Als Alternative zu QR-Codes können sich die digitalen Tickets bei der Kontrolle auch per Near Field Communication (NFC) ausweisen.

## Proprietär, aber transparent

Der Funktionsumfang hat Sie überzeugt und Sie wollen als Entwickler direkt loslegen, solche Tickets zu erzeugen? Dann haben wir zuerst ein paar nicht so gute Nachrichten: Das Format für die Wallet-Tickets hat sich Apple im stillen Kämmerlein ausgedacht. Es gibt also keinen (Web-)Standard und Apple kümmert sich auch nicht um den Teil der Nutzerschaft, der ohne iPhone aus dem Haus geht, sondern lieber ein Android-Gerät zum Konzert mitnimmt. Wie die Tickets dennoch aufs Android-Telefon gelangen, lesen Sie auf Seite 154.

### Und Android?

Auch auf dem Android-Telefon kann man die für Apples Betriebssysteme erstellten Tickets öffnen. Im Play Store gibt es die kostenlose App WalletPasses. Sie stammt von einem Entwickleraccount namens „WalletPasses Alliance“. Das klingt wie ein unabhängiger Herstellerverband, scheint aber ein Fantasiename zu sein. Außerhalb des Stores tritt ein Verband dieses Namens nirgends in Erscheinung. Auch Apple hat damit nichts zu tun. Die App funktioniert aber, kann Apples Datenformat lesen und zeigt die Passes nahezu im gleichen Layout wie die Original-App an.

Dass Apple in diesem Bereich jahrelang die Nase vorn hatte, schien auch Google zu stören. Auf der Entwicklerkonferenz Google I/O im Mai 2022 stellte das Unternehmen seine Pläne für Google Wallet vor. Nicht nur der Name erinnert an Apples App, auch das Konzept: Das Bezahlen mit Google Pay soll künftig nur eine Funktion der digitalen Brieftasche namens Google Wallet sein. Daneben sollen Kundenkarten, Tickets und Impfpässe ein digitales Zuhause bekommen. Die Entwicklerdokumentation für diese neuen Funktionen ist unter der Adresse [developers.google.com/wallet](https://developers.google.com/wallet) zu finden.

Nur weil das Format von Apple entwickelt wurde, bedeutet das zum Glück aber nicht, dass nur Apple-Software solche Tickets

erzeugen kann. Und anders als bei Transaktionen über den App-Store nimmt Apple auch keine Provision für jedes ausgestellte Ticket. Stattdessen ist das Format ausführlich dokumentiert (siehe [ct.de/yw3e](http://ct.de/yw3e)) und jeder mit einem Texteditor und einem Zip-Werkzeug könnte Tickets erzeugen. Es gibt jedoch einen Haken: Damit die Tickets, in Apples Terminologie Pass genannt, beim Kunden funktionieren, muss man sie digital signieren. Dafür braucht man ein Apple-Entwickler-Zertifikat, das man nur bekommt, wenn man sich als Entwickler im Apple-Developer-Programm registriert. Der Account berechtigt auch dazu, Apps in den Store zu bringen und damit Geld zu verdienen. 99 US-Dollar (und mit Steuern auch 99 Euro) kostet ein solcher Account für Einzelpersonen im Jahr. Wie Sie Apple-Entwickler werden und an ein Zertifikat zum Signieren der Tickets kommen, lesen Sie auf Seite 155.

## **Developer-Account und Zertifikat beschaffen**

Ein Apple-Developer-Account ist Voraussetzung, um Passes für die Wallet signieren zu können. Nebenbei dürfen Entwickler mit einem solchen Account auch Apps in den App Store bringen – 99 Euro kostet die Mitgliedschaft für ein Jahr. Um Mitglied zu werden, braucht man eine gewöhnliche Apple-ID, die man als normaler Nutzer zum Beispiel für die App-Stores und Apple Music nutzt. Der Login muss durch einen zweiten Faktor abgesichert sein. Los geht die Registrierung unter der Adresse [developer.apple.com/enroll](http://developer.apple.com/enroll). Der Assistent fragt zunächst nach der Art des Accounts – im einfachsten Fall meldet man sich als Privatperson an. Firmenangehörige, die mit dem Account auch Apps unter Firmennamen verkaufen wollen, müssen den anderen Knopf betätigen. Die Fragen des Assistenten sind weitestgehend schnell beantwortet – etwas verwirrend ist die Anforderung, die eigenen Kontaktdaten zweimal einzutippen, einmal davon „Romanized“. Das ist für asiatische Entwickler gedacht, die ihre Daten einmal in ihren Schriftzeichen und noch einmal im lateinischen Buchstabensystem eingeben sollen. Europäer füllen beide Abschnitte des Formulars einfach identisch aus.

Am Ende des Prozesses geht es ans Bezahlen per Kreditkarte, die PayPal-Option war bei uns ausgeblendet. 99 Euro und ein paar Minuten Wartezeit später bekommt man eine Mail und ist offiziell Apple-Entwickler. Unter der Adresse [developer.apple.com/account](https://developer.apple.com/account) findet man ab jetzt den internen Bereich.

Für jeden Typ von digitalen Tickets, aber nicht für jedes einzelne Ticket, brauchen Sie einen sogenannten Identifier, der im internen Bereich hinterlegt ist. Eine Fluggesellschaft bräuchte also für all ihre Flugtickets einen solchen Identifier. Wenn sie über die Wallet auch Essensgutscheine verteilen wollte, müsste sie einen neuen anlegen – und zu jedem Identifier braucht man ein von Apple signiertes Zertifikat. Damit kann man stets beliebig viele Tickets signieren.

## Certificates, Identifiers & Profiles

[← All Identifiers](#)

### Edit your Identifier Configuration Remove

Description	Identifier
ct demo pass	pass.de.heise.ct.demopass

#### Production Certificates

Name: Pass Type ID: Pass  
Type: Pass Type ID  
Expires: 2023/08/04

Revoke Download

Create an additional certificate to use for this Pass Type ID.

Create Certificate

Ein Identifier und ein passendes Zertifikat sind die Voraussetzung, um gültige Tickets für die Wallet zu erzeugen. Anlegen kann man sie im Developer-Portal mit einem kostenpflichtigen Account.

Klicken Sie zum Anlegen eines Identifiers im Entwicklerportal auf „Certificates, Identifiers & Profiles“ und dort unter

Identifiers auf das blaue Plus. Sie brauchen ein Objekt vom Typ „Pass Type IDs“. Anschließend geben Sie dem Identifier eine Beschreibung wie „Mein Demoticket“ und denken sich eine eindeutige Bezeichnung in Form einer umgekehrten Domain aus, wir haben zum Test pass.de.heise.ct.demopass gewählt. Das Formular ergänzt den ersten Block pass. automatisch, die Domain müssen Sie auch nicht besitzen – es geht nur darum, einen eindeutigen String zu generieren, den niemand sonst hat. Hat man den Assistenten durchgeklickt, liegt der Identifier bereit und kann mit einem Zertifikat verknüpft werden.

Klicken Sie den Eintrag in der Liste an und wählen dann „Create Certificate“. Der Assistent erwartet von Ihnen einen Namen (nur für Ihre eigene Sortierung) und den Upload eines CSR, eines „Certificate Signing Request“. Das ist eine Datei, die man auf dem lokalen Computer zusammen mit einem geheimen Schlüssel erzeugt. Der Schlüssel bleibt immer auf dem eigenen Gerät, der CSR ist eine schriftliche Bitte, den öffentlichen Schlüssel zu signieren. Als Antwort auf den CSR bekommen Sie von Apple einen signierten öffentlichen Schlüssel zurück.

Am schnellsten kommen Mac-Nutzer an einen passenden CSR (wen wundert es). Sie öffnen das Systemprogramm Schlüsselbundverwaltung und wählen oben links in der Menüleiste

„Schlüsselbundverwaltung/Zertifikatsassistent/Zertifikat einer Zertifizierungsinstanz anfordern ...“. Der Assistent hat nicht viele Fragen. Man gibt eine E-Mail-Adresse (wird nicht im Zertifikat eingebaut und ist nicht sichtbar) und einen Namen (Freitext, zum Beispiel „Wallet-Tickets“) ein. Das dritte Feld für die Mailadresse der Zertifizierungsstelle bleibt leer. Anschließend „Auf der Festplatte sichern“ anwählen, den Haken „Eigene Schlüsselpaarinformationen festlegen“ aktivieren und den Assistenten abschließen. Der CSR liegt dann im Dokumente-Ordner. Der private Schlüssel heißt Wallet-Tickets, liegt im Schlüsselbund und kann dort später exportiert werden, wenn man ihn braucht.

Unter Linux (und unter Windows im WSL) ist es ebenfalls möglich, an einen CSR zu kommen, wenn auch nicht von Apple dokumentiert. Führen Sie einfach folgenden Befehl aus:

```
openssl req -nodes -newkey rsa:2048 -keyout apple_pass.key -out CertificateSigningRequest.certSigningRequest
```

Die meisten Fragen des Assistenten können Sie überspringen, nur den Ländercode (DE) und die E-Mail-Adresse sollten Sie setzen. Auch bei diesem Verfahren landen die Daten nicht im Zertifikat und werden nicht angezeigt. OpenSSL erzeugt den CSR und den Schlüssel in der Datei `apple_pass.key`.

Unabhängig vom Betriebssystem schnappen Sie sich am Ende die erzeugte Datei mit dem sperrigen Namen `CertificateSigningRequest.certSigningRequest` und laden Sie, zurück im Developer-Portal, im Feld für den CSR hoch. Postwendend erhalten Sie eine Datei namens `pass.cer` mit Ihrem Zertifikat zurück.

## Datenstrukturkunde

Ein Pass ist technisch nur eine Zip-Datei mit ein paar verpflichtenden Inhalten. Nach dem Verpacken mit einem Zip-Programm muss man die Dateiendung `.zip` lediglich durch `.pkpass` ersetzen und kann sie anschließend verschicken. Um zu verstehen, wie die Tickets funktionieren, erfahren Sie zunächst, welche Dateien im Ordner liegen müssen – im Anschluss erhalten Sie Tipps, wie Sie das Generieren in bestehende Prozesse einbauen können. Denn obwohl man die Pässe theoretisch per Hand zusammenbauen bauen könnte, Spaß macht das nicht.

Alle Dateien, die zu einem Pass verpackt werden sollen, müssen in einem Ordner liegen. In diesen Ordner gehört verpflichtend eine Datei namens `pass.json`. Die enthält im JSON-Format alle Texte und Einstellungen. Zunächst sind da ein paar Pflichtfelder:

```
{
  "passTypeIdentifier":
    "pass.de.heise.ct.demopass",
  "serialNumber": "1234567890",
  "formatVersion": 1,
  "organizationName": "c't magazin"
  "description": "Einladung",
  "teamIdentifier": "<Ihre ID>",
  [...]
}
```

Der `passTypeIdentifier` ist die Zeichenkette, die Sie zuvor im Developer-Bereich angelegt haben. Die Seriennummer (`serialNumber`) müssen Sie als Aussteller generieren und sich darum kümmern, dass sie für alle Pässe mit einem Identifier eindeutig ist – mit einer Datenbank im Hintergrund, die Ihre Tickets verwaltet, sollte das kein Problem sein. Vorgaben zum Format gibt es nicht, viele von uns untersuchte Tickets großer Aussteller enthielten in diesem Feld UUIDs [1].

Die `formatVersion` ist schnell erklärt, dafür ist aktuell nur der Wert 1 zulässig. Die `description` soll laut Dokumentation keine Inhalte des Tickets (wie den Namen des Inhabers) enthalten. Es handelt sich um eine Beschreibung, die beispielsweise von den Assistenzwerkzeugen sehbehinderter Nutzer vorgelesen werden kann. Den `teamIdentifier` müssen Sie aus der Developer-Plattform kopieren. Sie finden ihn, indem Sie unter [developer.apple.com/account](https://developer.apple.com/account) links auf Membership klicken. Es handelt sich um eine zehnstellige Zeichenkette aus Großbuchstaben und Zahlen.

Es folgen auf der obersten Ebene des JSON-Objekts viele freiwillige Angaben, die man auch weglassen kann. Diese bestimmen unter anderen, wie das digitale Dokument aussehen soll. Ans eigene Farbschema passt man das Ticket mit `backgroundColor`, `foregroundColor` und `labelColor` an. Die Werte müssen wie in CSS als RGB-Farbe angegeben sein, andere Schreibweisen (wie die hexadezimale) sind nicht erlaubt. Ein Eintrag für ein leuchtendes Orange als Hintergrund sieht zum

Beispiel wie folgt aus:

```
"backgroundColor": "rgb(255,125,0)"
```

Außerdem kann man auf der obersten Ebene festlegen, wann das Dokument relevant ist, also beim Nutzer automatisch auf dem Sperrbildschirm auftauchen soll. Bei einem Konzertticket könnte das kurz vor dem Einlass sein. Wenn es eine eindeutige Zeit gibt, ist das der einfachste Weg, das Ticket automatisch in den Vordergrund zu rücken. Das Datumsformat muss dem W3C-Standard für Datum und Uhrzeit entsprechen, am einfachsten gibt man die Zeit in UTC an (signalisiert durch das Z am Ende):

```
"relevantDate":"2022-08-10T10:00Z"
```

Was für Eintrittskarten gut funktioniert, ist bei anderen Dokumenten gar nicht so nützlich. In einem Wallet-Pass kann zum Beispiel auch eine Kundenkarte stecken. Für solche hat sich Apple ein besonderes Mittel zur Kundenbindung ausgedacht – für datenschutzbewusste Europäer mag die aber übergriffig wirken, weshalb man sie sparsam einsetzen sollte. Im Pass kann man Geokoordinaten sowie eine Distanz hinterlegen. Nähert sich der Kunde einem Bereich um einen Laden, taucht der Pass mit einem Hinweistext auf. Dabei erhalten Sie als Aussteller keine Benachrichtigung und die Verarbeitung geschieht lokal auf dem Telefon – aber das weiß der datenschutzbewusste Kunde nicht und könnte sich verfolgt fühlen. Technisch funktioniert das wie folgt:

```
"locations": [  
  {  
    "latitude": 52.3859153,  
    "longitude": 9.80959388,  
    "relevantText": "Kommen Sie herein!"  
  }  
],  
"maxDistance": 100
```

Unter locations können Sie bis zu zehn Orte mit ihren

Koordinaten hinterlegen, die `maxDistance` ist der Abstand in Metern. Unterschreitet das Telefon diesen Abstand, erscheint der Pass mit dem unter `relevantText` angegebenen Werbetext. Haben Sie zu viele Geschäfte, um sie alle unter `locations` einzutragen, gibt es eine Alternative: Sie können auch Bluetooth-Beacons in den Läden aufhängen und die digitalen Tickets darauf reagieren lassen – die Dokumentation (siehe [ct.de/yw3e](https://ct.de/yw3e)) verrät, welche Datenfelder dafür vorgesehen sind. Wer es mit solchen Kundenbindungsmaßnahmen übertreibt, muss damit rechnen, dass die Kunden die virtuelle Kundenkarte schnell wieder aus der Wallet-App werfen.

Das Rauswerfen funktioniert aber nicht nur manuell, wenn der Nutzer vom Pass genervt ist, sondern auch automatisch, wenn ein Ticket nicht mehr aktuell ist. Gerade Veranstaltungs- und Reisetickets möchte man am Tag darauf nicht mehr in der App sehen (oder höchstens in einer Art Papierkorb). Mit einem `expirationDate` (im W3C-Datumsformat wie auch das `relevantDate`) nehmen Sie den Kunden das Aufräumen ab.

## Karten für alle Gelegenheiten

Nach diesen verpflichtenden und freiwilligen Angaben müssen Sie sich entscheiden, um welchen Typ digitales Dokument es sich handelt und diesen in der Datei `pass.json` eintragen (mehr dazu später). Zur Auswahl stehen:

- `eventTicket`: Eintrittskarte zu einer Veranstaltung
- `boardingPass`: Ticket für Flüge, Bus- oder Bahnreisen
- `coupon`: Gutschein oder Rabattcoupon
- `storeCard`: Kunden- oder Mitgliedskarte
- `generic`: sonstige digitale Karten

Für Verkehrsunternehmen, Veranstaltungsorganisatoren, Ladenbetreiber und Fitnessstudios gibt es somit passende Designvorlagen, für alle anderen Dokumente ist `generic` gedacht. Mit der Wahl eines Typs entscheidet man sich für ein Layout, im Prinzip funktionieren aber alle Typen identisch und

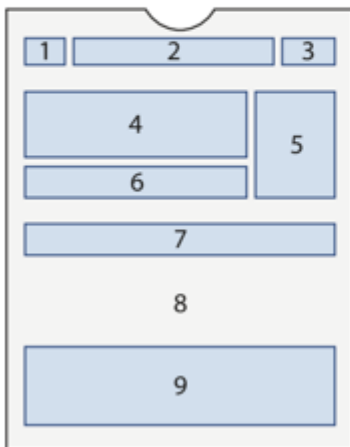
unterscheiden sich bei den Funktionen nur in Nuancen. Als Ersteller darf man vier Bereiche mit Texten füllen – in der Infografik auf Seite 156 haben wir die Abschnitte markiert und beschriftet. Im Bereich oben rechts (headerFields) kann man bei einer Veranstaltung zum Beispiel das Datum gut sichtbar hinterlegen. Die Wallet-App simuliert ja eine Briefftasche und stapelt mehrere Tickets hintereinander. Diese Kopfzeile ist immer lesbar und wird nicht von anderen Karten verdeckt.

Die zentralen Informationen, die groß und fett geschrieben werden sollen, liegen in primaryFields und erstrecken sich oben quer über das Ticket. Die nächste Ebene (kleiner dargestellt und mit mehreren Einträgen nebeneinander) heißt secondaryFields, noch kleiner werden auxiliaryFields dargestellt. Die virtuelle Variante der Ticketrückseite heißt bei Apple backFields und ist gedacht für Geschäftsbedingungen und anderes Kleingedrucktes. Angezeigt werden die Daten nur, wenn der Nutzer die drei Punkte oben rechts auf einem Ticket antippt.

Apple hat sich dagegen entschieden, Datenmodelle für allerlei Gelegenheiten mit starren Feldern zu spezifizieren (etwa Datum, Sitzplatz, Start, Ziel, Reisender, Gate, Bahnsteig). Stattdessen kann man in den Bereichen alle Angaben in Schlüssel-Werte-Paaren frei definieren und statt eines Konzerttickets auch eine digitale Kantinenkarte bauen.

# Eintrittskarten für Apple Wallet

Der Aufbau der Tickets ist von Apple vorgegeben, innerhalb der Felder kann man sich mit Informationen ausbreiten und außer Texten auch Bilder und QR- oder Barcodes platzieren.



- 1 Logo (logo.png)
- 2 Text neben dem Logo (logoText)
- 3 Kopfzeile (headerFields)
- 4 Zentrale Informationen (primaryFields)
- 5 Zusätzliches Bild (thumbnail.png)
- 6 Weitere Informationen (secondaryFields)
- 7 Noch mehr Informationen (auxiliaryFields)
- 8 Hintergrundbild (background.png)
- 9 QR- oder Barcodes (barcodes)

Zunächst legt man den gewählten Typ im JSON-Dokument auf der obersten Ebene als Objekt an, darunter Objekte für die Felder, die man nutzen möchte (primaryFields, secondaryFields ...). Sie sind allesamt optional. In den Feldern platziert man seine Inhalte jeweils mit einem Schlüssel (key), einer Beschriftung (label) und dem Wert (value). Das Label steht dann im fertigen Pass über oder unter dem Wert.

Bei einem Veranstaltungsticket für eine fiktive Grillfeier mit Freunden kann das zum Beispiel folgendermaßen aussehen:

```
"eventTicket": {
  "headerFields": [
    {
      "key": "seat",
      "label": "SITZPLATZ",
      "value": "Bierbank 1"
    }
  ],
  "primaryFields": [
    {
      "key": "event",
      "label": "VERANSTALTUNG",
      "value": "Grillfeier mit Freunden"
    }
  ]
}
```

```

],
"secondaryFields": [
  {
    "key": "location",
    "label": "ORT",
    "value": "Garten"
  }
],
"auxiliaryFields": [
  {
    "key": "food",
    "label": "MITBRINGEN",
    "value": "Nudelsalat"
  }
],
"backFields": [
  {
    "key": "terms",
    "label": "Bedingungen",
    "value": "Bitte bis 17:00 absagen"
  }
]
}

```

Bei der Vergabe der Schlüsselnamen ist man frei, sie müssen nur einmalig sein. Falls Sie zufällig in die Verlegenheit kommen sollten, Tickets für ein Reiseunternehmen zu programmieren, gibt es noch eine Besonderheit. Sobald Sie das Objekt `boardingPass` angelegt haben, müssen Sie darin verpflichtend einen `transitType` angeben. Im Ticket erscheint dann zwischen den beiden primären Informationen in `primaryFields` ein passendes Icon. Ein Ausschnitt aus einem Flugticket kann zum Beispiel wie folgt aussehen:

```

"boardingPass": {
  "transitType": "PKTransitTypeAir",
  "primaryFields": [
    {
      "key": "from",
      "label": "VON",
      "value": "HAJ"
    }
  ]
}

```

```

    },
    {
      "key": "to",
      "label": "NACH",
      "value": "LYR"
    }
  ]
}

```

Die Buchstaben PK stehen für PassKit, so heißt das Framework für die Tickets und auch für den Zugriff auf Apple Pay. Neben TypeAir gibt es noch TypeBoat, TypeBus, TypeGeneric und TypeTrain.

## Codes und Bilder

Sind all diese Werte ausgefüllt, gibt es noch zwei weitere optionale Aufgaben. Auf der einen Seite sind das die Bilddateien für Logos. Dabei kann (muss aber nicht) sich ein Grafiker genauso austoben wie bei den Favicons für Websites und noch Varianten für unterschiedlich große Bildschirme erzeugen (mit @2x am Ende des Dateinamens). Die Arbeit fällt pro Unternehmen nur einmal an. In der Tabelle auf Seite 158 sehen Sie, welche Bildchen Sie anlegen und in den Ordner neben die pass.json legen können. Alle Bildchen sind optional.

Letzte Baustelle ist der Code, mit dem Sie die Echtheit des Dokuments prüfen. Das kann ein Barcode sein, ein QR-Code oder auch die NFC-Schnittstelle des Mobiltelefons. Für letztere Technik ist zusätzlicher Zertifikatsaufwand nötig, dazu sei auf die Dokumentation verwiesen (siehe [ct.de/yw3e](http://ct.de/yw3e)). Bar- und QR-Codes dagegen sind schnell eingebunden und gehören auf die oberste Ebene im JSON-Objekt. Sie sind wie die meisten Felder optional:

```

barcodes": [
  {
    "altText": "123 Mustermann",
    "format": "PKBarcodeFormat",
    "messageEncoding": "iso-8859-1",

```

```
"message": "123MUSTERMANN"  
}  
]
```

Die Wahl des Formats hängt vor allem von Ihrer bestehenden Leser-Infrastruktur ab. In einen QR-Code passen die meisten Daten, neben PKBarcodeFormatQR sind auch FormatPDF417 und FormatAztec vorgesehen. Apple verweist darauf, dass man heutzutage unter dem Schlüssel barcodes eine Liste mit potenziell mehreren Einträgen anlegen soll. Ab iOS 9.0 wird diese Schreibweise verwendet. Früher legte man einen einzelnen Barcode unter barcode ab. Das ist seit dem Start von iOS 9.0 offiziell abgekündigt und wäre nur noch für iOS 8 und noch ältere Systeme nötig. Bei unseren Tests fanden wir etwa bei einem Ticket der Lufthansa aus dem Jahr 2022 ausschließlich den alten Eintrag barcode im Singular.

Bilddateien für Logos und Hintergründe		
Dateiname	Zweck	Format <sup>1</sup>
background.png	Großflächiges Hintergrundbild	180 × 220
footer.png	Kleiner Streifen über dem Barcode	286 × 15

Bilddateien für Logos und Hintergründe		
Dateiname	Zweck	Format <sup>1</sup>
icon.png	Quadratisches Logo, wenn das Ticket auf dem Home-Screen angezeigt wird	29 × 29
logo.png	Logo des Unternehmens, wird oben links neben dem headerFields angezeigt	160 × 50 (oder schmaler)
strip.png	Zusätzlicher Hintergrund für den Bereich primaryFields	375 × 123
thumbnail.png	Zusätzliches Logo, wird bei Veranstaltungsticketes auf der Vorderseite rechts dargestellt	90 × 90
<sup>1</sup> In Pixeln. Zusätzlich kann man alle Bilder für hochauflösende Bildschirme noch in hochskalierten Varianten anlegen und die Dateinamen mit @2x.png kennzeichnen.		

## Zur Prüfung

Die Datei pass.json und die Bildchen liegen im Ordner, damit ist das Ticket fast fertig zum Verpacken. Eine letzte verpflichtende Datei fehlt aber noch. Sie heißt manifest.json und dient der Signaturprüfung. Dafür enthält sie ein Inhaltsverzeichnis aller Dateien im Ordner im JSON-Format. Stark gekürzt sieht sie zum Beispiel so aus:

```

{
  "pass.json": "74342dc0649fc1 [...]",
  "icon.png": "e0f0bcd503f611 [...]",
  "logo.png": "66a0989dcf0c5c [...]",
  [...]
}

```

Jede Datei, die zum Pass gehört, bekommt hier einen Eintrag mit ihrem Dateinamen und dem SHA1-Hash ihres Inhalts. Die Idee dahinter: Beim Verpacken erzeugt der Ersteller mithilfe seines Apple-Zertifikats für diese Manifest-Datei eine Signatur im Format „PKCS #7 Detached“, das zum Beispiel auch bei E-Mails mit S/MIME zum Einsatz kommt. Die erzeugte Signatur legt er mit dem Dateinamen signature mit in den Ordner. Beim Laden vollzieht die Wallet-App auf dem Telefon diese Schritte zum Test einmal nach und bildet ebenfalls alle Hash-Werte aller Dateien. Hätte zwischendurch jemand eine der Dateien auch nur um ein Bit manipuliert, wäre schlagartig ihr SHA1-Hash ungültig, dadurch die Manifest-Datei und schließlich auch die Signatur.



Auch Reisetickets und Kundenkarten finden in der Wallet-App

ihren Platz.

Den gesamten Signaturprozess darf man aber nicht überbewerten – er stellt letztlich nur sicher, dass ein registrierter Entwickler, der den `passTypeIdentifier` bei Apple angelegt hat, das Ticket mit seinem Schlüssel signiert hat. Am Einlass beim Konzert oder am Flughafen sagt das aber überhaupt nichts aus, den Identifier sieht das Kontrollpersonal nicht einmal und die Authentifizierung erfolgt lediglich über den QR- oder Barcode. Jeder mit Apple-Account kann Wallet-Pässe basteln und signieren, die wie Bahn- oder Lufthansa-Tickets aussehen. Herausfinden kann man den wahren `passTypeIdentifier` nur, wenn man den Pass in Ruhe auf einem PC untersucht.

Daraus folgt auch ein ganz legaler Praxistipp für Entwickler, die eigene Pässe bauen und sich inspirieren lassen wollen: Sie sollten einen Pass eines bekannten Anbieters auf dem Telefon öffnen, oben rechts auf die drei Punkte klicken und ihn sich per AirDrop oder Mail an den PC schicken. Dort ändert man einfach die Dateiendung `.pkpass` in `.zip` und entpackt das Archiv. Auf diese Weise lässt sich erfahren, wie die großen Anbieter ihre Pässe bauen.

## Vom Fließband

Mit diesen Erklärungen und etwas Recherche in bestehenden Tickets sind Sie in der Lage, eigene Tickets und Karten für viele Gelegenheiten zu gestalten. Erklärungen zu allen Funktionen, die Apple sonst noch in den Passes versteckt hat, finden Sie in der Dokumentation über [ct.de/yw3e](https://ct.de/yw3e). Nützlich für einige Szenarien, wenn auch nicht mal eben eingerichtet, sind die Themen Mehrsprachigkeit und Updates über einen Server – wie Sie letztere Funktion zum Laufen bringen, würde den Umfang dieses Artikels deutlich sprengen. Bevor Sie sich an solche Herausforderungen machen, sollten Sie Ihren ersten gültigen und signierten Pass erzeugen.

Wir empfehlen dringend, den Signaturprozess mit der Manifest-Datei nicht per Hand durchzuspielen. Die Debugging-

Möglichkeiten sind auf den ersten Blick begrenzt. Entweder ein Ticket ist gültig oder das Telefon weigert sich mit einer nichtssagenden Fehlermeldung. Wer ohnehin schon im Apple-Universum entwickelt, kommt mit dem iOS-Simulator von XCode immerhin an detailliertere Fehlermeldungen im System-Log.

Machen Sie sich das Leben einfacher und suchen Sie nach einer Apple-Wallet-Bibliothek in der Programmiersprache Ihres Vertrauens und erzeugen Pässe lieber programmatisch. Die Bibliotheken funktionieren alle ähnlich: Sie erwarten den privaten Schlüssel und das Zertifikat. Im Programmcode hinterlegt man statische Werte, die bei allen Tickets identisch sind, als Vorlage. Die dynamischen Inhalte (ID, Reisedatum, Name des Reisenden ...) werden dann zur Laufzeit eingebacken. Heraus fällt eine verpackte und signierte pkpass-Datei.

Über [ct.de/yw3e](https://ct.de/yw3e) finden Sie eine Sammlung mit Bibliotheken für verschiedene Programmiersprachen. Für private Experimente kann man anhand der Dokumentationen und mit Kenntnis der jeweiligen Sprache schnell einen Prototypen zusammenstricken. Im Unternehmenseinsatz ist das Generieren solcher Tickets ein Musterbeispiel für den Einsatz eines Microservices – zum Beispiel in Form eines kleinen Containers, der intern ein HTTP-API anbietet und von einem anderen System (das Buchungen verarbeitet) die variablen Ticket-Daten als JSON-Daten annimmt. Zurück liefert der Microservice einen fertigen Pass. Wenn Sie sich für eine solche Umsetzung interessieren, werfen Sie mal einen Blick auf unser Repository zum Artikel – da haben wir einen solchen Microservice als Docker-Container zusammengebaut. Übergeben müssen Sie dem Container nur Ihr Zertifikat und Ihren privaten Schlüssel.

## Fazit

Mit seiner Wallet-App und dem eigenen Dateiformat hat Apple ein echtes Alltagsproblem gut und recht flexibel gelöst. Leider gilt dasselbe wie bei Apps in der Mobilgeräte-Welt:

Möchte man die Android-Nutzer nicht ausschließen, muss man wohl oder übel verschiedene Pässe erzeugen. Apple und Google gehen in diesem Bereich getrennte Wege, statt Standards zu vereinbaren.

Hat man die Hürde mit Entwickler-Account und Zertifikat überwunden, ist der Rest der Arbeit einfaches JSON-Handwerk. Beim Datenformat gilt dankenswerterweise: Vieles kann, wenig muss. Um ein Grundgerüst zu einem Pass zu verschnüren, sollte man dann zu einer Softwarebibliothek in der Sprache des Vertrauens greifen. So steht eigenen digitalen Tickets nichts mehr im Weg. ([jam@ct.de](mailto:jam@ct.de))

1. Literatur
2. [Jan Mahn, Zufall schlägt das System, Ein Plädoyer für UUIDs in Datenbanken, c't 6/2022, S. 138](#)

Dokumentation und Beispiel-Container: [ct.de/yw3e](https://ct.de/yw3e)

---

# Texte auszeichnen mit Markdown

## # Überschrift

# Mit Markdown schnell und einfach Texte auszeichnen

Ob Blog-Einträge, Kommentare in Foren oder Bugtracker auf Entwicklungsplattformen – immer mehr Software unterstützt Markdown zum Auszeichnen von Texten. Nach unserer Einführung

wissen Sie nicht nur, warum der Überschrift ein # voransteht, sondern auch, wie Sie Textpassagen kursiv stellen, Aufzählungen und Tabellen eintippen und vieles mehr.

Von Sylvester Tremmel

## **kompakt**

- GitHub, Reddit, Trello – zahllose Systeme verstehen Markdown; unsere Einführung erklärt die weit verbreitete Auszeichnungssprache.
- Markdown zu schreiben geht schnell und fällt leicht; wer die Sprache beherrscht, nutzt sie oft auch für die eigenen Texte und Notizen.
- Auch in kollaborativen Projekten wird dadurch niemand ausgeschlossen, lesbar sind Markdown-Texte ohne Kenntnis der Sprache und in jedem beliebigen Texteditor.

Mit Markdown ergänzt man reinen Text leicht um Formatierungsinformationen, die eine visuell schöne Darstellung erlauben. Der Clou: Markdown-Texte sind nicht nur bequem zu schreiben, sondern auch im Quelltext gut lesbar. Dadurch benötigt man keine Textverarbeitung wie Microsoft Word oder LibreOffice Writer. Jeder beliebige Texteditor eignet sich, um Markdown zu verfassen und zu lesen. Die schöne Darstellung durch passende Software ist ein Bonus fürs Endergebnis, keine Voraussetzung beim Arbeiten an den Texten.

Weil Markdown so praktisch ist, kennen mittlerweile alle möglichen Systeme die Sprache: zahlreiche Notiz-Apps wie etwa „Drafts“ [1] oder „Obsidian“ [2]; komplexe Systeme zum Schreiben von Texten wie „Notion“ [3] oder „HedgeDoc“ [4] und Nischenanwendungen wie die Rezeptdatenbank „Tandoor Recipes“ [5]. Große Serveranwendungen wie GitHub oder GitLab und viele Systeme für Internetforen erlauben ebenfalls, die eigenen Beiträge und Nachrichten mit Markdown auszuzeichnen. Und in seinem ursprünglichen Zweck – dem Schreiben von Texten zur Publikation im Internet – glänzt Markdown ebenfalls: Zahllose

Blogging-Systeme, Website-Editoren [6] und Dokumentations-Generatoren unterstützen die Sprache.

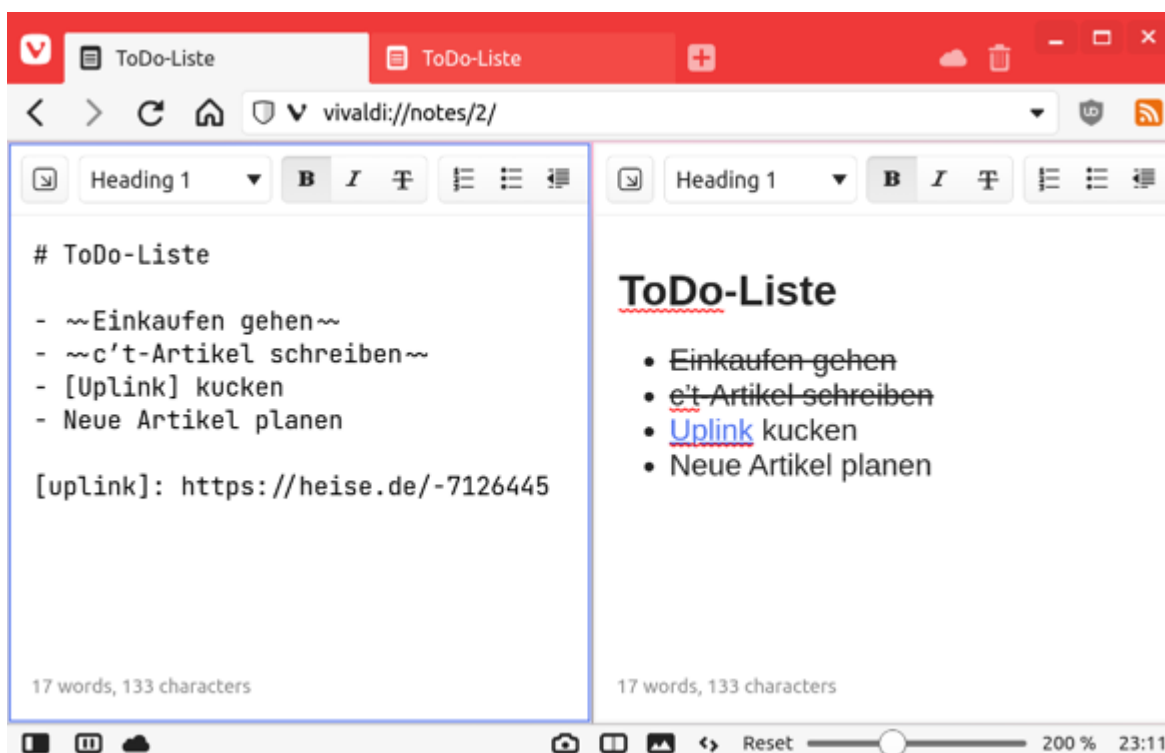
Die weite Verbreitung hat leider den Nebeneffekt, dass diverse Varianten der Sprache existieren (siehe Kasten). Die grundlegenden Features funktionieren aber überall gleich (oder zumindest sehr ähnlich) und was diese Einführung zeigt, sollte in den meisten Markdown-Implementierungen problemlos funktionieren. Der Text orientiert sich grob an CommonMark, weist aber auch darauf hin, wenn es andernorts nette Erweiterungen gibt oder etwas nicht ganz gleich funktioniert.

## **Pandoc's GitHub-flavoured CommonMark?**

Nicht überall, wo „Markdown“ draufsteht, ist das gleiche Markdown drin. Die ursprüngliche Version der Sprache wurde von John Gruber und Aaron Swartz aus der Taufe gehoben, um damit Artikel fürs Web zu schreiben. Mittlerweile findet Markdown allerdings in allen möglichen Systemen Verwendung und die immer neuen Einsatzzwecke luden zu allerlei Erweiterungen der Sprache ein. Hinzu kam, dass Grubers Dokumentation der Markdown-Syntax (alle Links unter [ct.de/y5hr](http://ct.de/y5hr)) an einigen Stellen nicht ganz eindeutig ist oder von seiner Referenzimplementierung abwich.

Manche der entstandenen Varianten sind kaum dokumentierte Ad-hoc-Ideen von den Autoren der jeweiligen Software, andere – wie etwa „GitHub Flavoured Markdown“ oder „Pandoc's Markdown“ – sind weit verbreitete und sauber spezifizierte Erweiterungen der Sprache. Den Wildwuchs wieder einhegen soll „CommonMark“, eine erweiterte und präzisierete Markdown-Spezifikation. Das funktioniert nicht ganz: Die CommonMark-Spezifikation sollte möglichst kompatibel zu bereits existierenden Praktiken sein und geriet daher teilweise recht komplex. Infolgedessen weichen verschiedene Markdown-Implementierungen in Details von CommonMark und auch voneinander ab. Außerdem gibt es weiterhin viele Features, die über CommonMark hinausgehen und von System zu System unterschiedlich funktionieren.

Ungeachtet dieser Einschränkungen ist CommonMark viel wert, weil es eine relativ verlässliche Basis darstellt: Wer die dort spezifizierten Features kennt und nicht auf abstruse Weise miteinander kombiniert, der hat mit jeder vernünftigen Markdown-Implementierung Freude. Und wenn ab und an eine Kleinigkeit auf dem jeweiligen System nicht funktioniert, weiß der erfahrene Markdown-Autor, das Problem zu umschiffen. Je nach Anwendungsfall und Bedarf kann man dann immer noch nachschlagen, welche über CommonMark hinausgehenden Möglichkeiten die jeweilige Software bietet.



Markdown kann man an den verschiedensten Stellen nutzen, hier verschönert es den Inhalt der Notizen-Funktion im Browser Vivaldi.

## Jeder Text ist Markdown

Markdown zu schreiben fällt grundsätzlich leicht, weil jeder Text Markdown ist – ungültige Syntax gibt es nicht. Wenn ein Markdown-Textabschnitt nicht anderweitig interpretiert werden kann, dann ist er eben genau das: ein Textabschnitt, genauer gesagt ein Absatz:

Ein Absatz in Markdown

Und noch einer, der sich über zwei Zeilen erstreckt.

Einfache Zeilenumbrüche behandelt Markdown wie Leerzeichen, sodass man den Quelltext auch manuell umbrechen kann – wie beim zweiten Absatz in diesem Beispiel –, ohne dass diese Umbrüche die Ausgabe beeinflussen. Um einen neuen Absatz zu beginnen – der auch in der Ausgabe als neuer Absatz gerendert wird –, muss man eine Leerzeile einfügen – was auch im Texteditor deutlicher ist.

Um einen Zeilenumbruch auch in der Ausgabe zu produzieren, reicht es, im Markdown-Text die vorhergehende Zeile mit einem Backslash oder mehr als einem Leerzeichen zu beenden:

Ein Absatz mit einem\  
Zeilenumbruch und `\`  
noch einem.

Um einen neuen Textabschnitt einzuleiten, schreibt man drei Sternchen (\*\*\*) , Bindestriche (---) oder Unterstriche (\_\_\_). Markdown interpretiert dergleichen als „thematischen Bruch“, den die meisten Systeme als horizontale Linie rendern. Wer mag, darf auch mehr Zeichen setzen oder sie mit Leerzeichen auflockern:

\*\*\*\*\*

- - - -

## Zweierlei Überschriften

Um eine Textzeile als Überschrift zu markieren, stellt man ihr ein oder mehrere Doppelkreuze (#) und ein Leerzeichen voran. Die Anzahl der Doppelkreuze (maximal sechs) gibt die Ebene der Überschrift an:

# Überschrift

## Unterüberschrift

Überschriften dürfen auch mit Doppelkreuzen enden, was manche Autoren schöner finden:

```
### Unterunterüberschrift ###
```

Markdown ist das egal, es ignoriert Doppelkreuze am Ende einfach und achtet auch nicht darauf, ob ihre Anzahl zur Zahl der öffnenden Zeichen passt.

Übrigens hilft ein Backslash, wenn Markdown ein Zeichen interpretiert, das einfach Textinhalt sein soll:

```
\# Normaler Text, beginnt mit #
```

```
# Überschrift, endet mit \#
```

Daneben kennt Markdown noch eine zweite, sehr augenfällige Syntax für Überschriften:

```
Überschrift  
=====
```

```
Unterüberschrift  
-----
```

Solche Überschriften nennt man Setext-Überschriften, nach einem Textauszeichnungssystem von 1991, aus dem die Syntax übernommen wurde. Setext-Überschriften sehen zwar im Texteditor schön aus, sind aber eher mühsam zu tippen. Markdown begnügt sich zwar auch mit nur einem einzelnen Gleichheitszeichen oder Bindestrich (oder jeder anderen Anzahl), allerdings geht dann schnell die bessere Optik verloren. In jedem Fall kann man mit dieser Syntax nur zwei Überschrift-Ebenen auszeichnen.

## Syntax für Faule

Bei einem anderen Feature lehnt sich die Markdown-Syntax ebenfalls an Bekanntes an: Zitate leitet man – wie in E-Mails – mit Größer-als-Zeichen ein. Auch das Verschachteln funktioniert:

```
> Zitat, das sich über  
> mehrere Zeilen  
> erstreckt.  
>  
>> Unter-Zitat mit  
>> zwei Zeilen und einer  
>>  
>> # Überschrift
```

Die Leerzeilen vor dem (Unter-)Zitat und der Überschrift dürfen laut CommonMark-Standard entfallen, aber manch andere Markdown-Implementierung besteht darauf. Grundsätzlich hilft es bei unerwarteten Ergebnissen oft, Elemente mit Leerzeilen zu trennen.

Wer seinen Text manuell umbricht, muss allerdings nicht jeder Zeile einzeln ein > voranstellen. „Markdown erlaubt Dir faul zu sein“, steht schon in der originalen Dokumentation von John Gruber. Bei Zeilen, die nur den aktuellen Absatz fortsetzen, darf man auf die Markierung verzichten. Das folgende Codebeispiel hat daher die gleiche Bedeutung wie das vorhergehende:

```
> Zitat, das sich über  
mehrere Zeilen  
erstreckt.  
>  
>> Unter-Zitat mit  
zwei Zeilen und einer  
>>  
>> # Überschrift
```

Ebenfalls kaum gewöhnungsbedürftig – und ebenfalls mit einer eingebauten Abkürzung für Faule – ist die Art und Weise, wie Markdown Listen auszeichnet. Man stellt dem Text für einen Listeneintrag schlicht einen Listenmarker voran. Die Sprache erlaubt dafür Sternchen, Bindestriche oder Plus-Zeichen für unnummerierte Listen und Zahlen mit Punkten für nummerierte Listen:

- \* Listenpunkt
- \* noch ein Listenpunkt
- \* und ein weiterer

1. Eins
2. Zwei
3. Drei

Statt dem Punkt nach einer Zahl darf man auch eine Klammer setzen (1), 2), ...). Übrigens beachtet Markdown nur die erste Zahl einer nummerierten Liste. Man kann also auch drei Punkte mit 3., 10. und 07. nummerieren und erhält als Ausgabe eine Liste, die 3., 4., 5. hochzählt. Das hilft, wenn Listenpunkte immer wieder an neue Positionen wandern: Man nummeriert einfach jeden Punkt mit 1. und Markdown kümmert sich darum, dass am Ende eine ordentliche Aufzählung herauskommt.

Andere Zählvarianten kennt CommonMark nicht, aber viele Markdown-Implementierungen erlauben auch römische Zahlen (i., ii., ...), Buchstaben (A), B), ...) oder vollständige Umklammerungen ((a), (b), ...). Was funktioniert, kann man einfach von Fall zu Fall ausprobieren.

Eine ebenfalls häufig verfügbare Ergänzung zu CommonMark sind Aufgabenlisten. Dafür folgen auf den Listenmarker zwei eckige Klammern, die ein Kästchen andeuten. Erledigte Einträge erhalten ein X in ihrem Kästchen:

- [ ] Aufgabe 1
- [x] Aufgabe 2
- [ ] Aufgabe 3

Bei der Ausgabe werden dann statt der Klammern ordentliche Kästchen und Häkchen angezeigt.

Wenn ein Listeneintrag mehrere Zeilen umfassen soll, dann rückt man die Folgezeilen einfach passend ein. Über Einrückungen werden Listen auch verschachtelt:

- + Listenpunkt und
- + ein langer Listenpunkt,

- der über mehrere  
Zeilen geht
- + Noch ein Punkt
  1. Ein Unterpunkt
  2. und noch einer
    - > mit einem Zitat

im Listenpunkt

Wie erwähnt ist die Leerzeile vor dem Zitat laut Standard optional, wird aber von so manchem System verlangt. Die Leerzeile danach ist in jedem Fall nötig, weil Markdown sonst eine „faule“ Zitatfortsetzung erkennt.

Die genauen Regeln für die Tiefe von Einrückungen sind aus mehreren Gründen sehr kompliziert. Auf der sicheren Seite bleibt, wer Einrückungen so wählt, dass alle Marker einer Liste untereinander liegen und auch der Text jeder Zeile auf Linie beginnt – also so, wie es oben beim zweiten Listenpunkt und zweiten Unterpunkt der Fall ist.

Wie erwähnt ist auch bei Listenpunkten Faulheit gestattet und man kann die Einrückung komplett weglassen. Das funktioniert aber nicht für Überschriften und dergleichen, sondern nur, wenn eine Zeile schlichten Text aus der Zeile darüber fortsetzt:

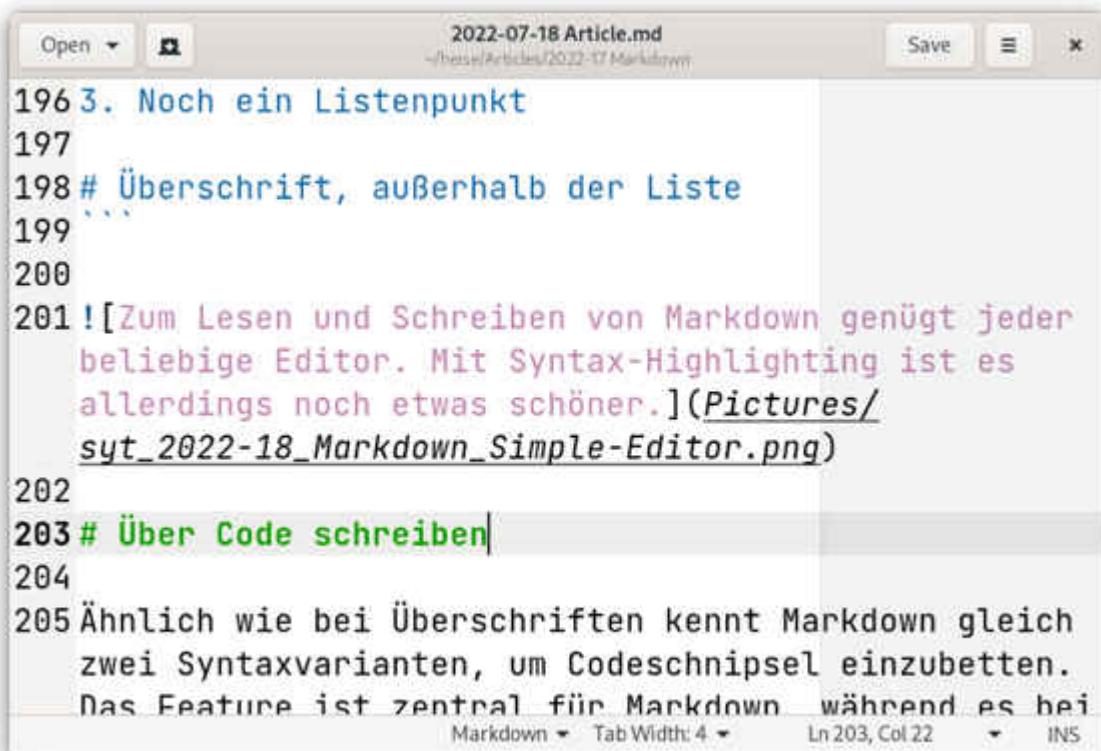
1. Ein langer Listenpunkt,  
über zwei Zeilen.

2. Ein Listenpunkt mit

# integrierter Überschrift

3. Noch ein Listenpunkt

# Überschrift, außerhalb der Liste



```
2022-07-18 Article.md
~/hese/Articles/2022-17 Markdown

196 3. Noch ein Listenpunkt
197
198 # Überschrift, außerhalb der Liste
199 ...
200
201 ![Zum Lesen und Schreiben von Markdown genügt jeder beliebige Editor. Mit Syntax-Highlighting ist es allerdings noch etwas schöner.](Pictures/syt_2022-18_Markdown_Simple-Editor.png)
202
203 # Über Code schreiben
204
205 Ähnlich wie bei Überschriften kennt Markdown gleich zwei Syntaxvarianten, um Codeschnipsel einzubetten. Das Feature ist zentral für Markdown während es bei
```

Zum Lesen und Schreiben von Markdown genügt jeder beliebige Editor. Mit Syntax-Highlighting ist es allerdings noch etwas schöner.

## Über Code schreiben

Ähnlich wie bei Überschriften kennt Markdown gleich zwei Syntaxvarianten, um Codeschnipsel einzubetten. Das Feature ist zentral für Markdown, während es bei anderen Textverarbeitungen und Auszeichnungssprachen eher ein Nischendasein fristet. Die Sprache kommt häufig in der Softwareentwicklung zum Einsatz und Programmierer müssen eben allenthalben Code dokumentieren, kommentieren oder publizieren.

In der ersten und simpleren Variante rückt man Codeschnipsel einfach vier Leerzeichen weit ein. Den Inhalt eines Codeblocks fasst Markdown nicht an, sodass man dort nach Herzenslust #, \* und alle anderen Zeichen verwenden kann:

```
Codezeile
  Noch eine Codezeile
```

```
# Keine Überschrift, sondern
# ein Kommentar im Code
```

Eingerückte Code-Blöcke sind der Grund, warum man die meisten anderen Markdown-Elemente (wie Zitate, Überschriften et cetera) zwar mit Leerzeichen einrücken darf, wenn man will, aber höchstens drei Leerzeichen weit. Rückt man ein Element vier Leerzeichen weit ein, macht Markdown daraus einen Codeblock.

Überhaupt wird es schnell unübersichtlich, wenn man eingerückte Codeblöcke mit Elementen wie Listenelementen kombiniert, die ebenfalls Einrückungen erfordern. Man kann solche Codeblöcke durchaus in Listen verschachteln und mit anderen Listeninhalten mischen, aber beim Schreiben artet das rasch in mühsames Leerzeichen-Abzählen aus.

Hier hilft die zweite Syntax für Codeschnipsel, die Codeblöcke nicht einrückt, sondern oben und unten mit expliziten Zeichen umzäunt („fenced code blocks“). So ein Zaun besteht aus drei oder mehr Gravis (`), oft auch „Backticks“ genannt) oder Tilden (~):

```
```
Codezeile
Noch eine Codezeile
```

```
# Keine Überschrift, sondern
# ein Kommentar im Code
```
```

Der abschließende Zaun muss mindestens so lang sein wie der öffnende. In aller Regel schreibt man sie gleich lang; diese Regel dient aber dazu, Codezeilen mit zum Beispiel drei Gravis im Codeblock zu erlauben, ohne dass sie fälschlicherweise den Codeblock abschließen: Man nutzt dann einfach vier oder mehr Gravis als Zaun – oder eben Tilden.

Umzäunte Codeschnipsel haben noch einen weiteren Vorteil gegenüber ihren eingerückten Pendanten: Sie ermöglichen

sogenannte Info-Strings. Das sind Metadaten zum Codeblock, die man direkt nach dem öffnenden Zaun in dieselbe Zeile schreibt. Der Inhalt dieser Metadaten hängt vom konkreten System ab, für das der Markdown-Text verfasst wird. In aller Regel ist das erste (und oft einzige) Wort der Name der Programmiersprache, die im Codeschnipsel zum Einsatz kommt. Das dient zum Beispiel dazu, ein passendes Syntax-Highlighting zu aktivieren:

```
```javascript
alert('Hello World!');
```
```

Wer statt einem ganzen Codeblock nur schnell ein paar Wörter im Fließtext als Code markieren will, kann sie einfach links und rechts in je einen oder mehrere Gravis einschließen. Tilden funktionieren dafür nicht. Wenn der markierte Code selbst eine Folge von Gravis enthält, dann nutzt man zum Umfassen schlicht eine Anzahl an Zeichen, die nicht im Code als Sequenz auftritt (und trennt sie, falls nötig, mit Leerzeichen vom Inhalt):

Ein Text mit `Code`,  
etwas ``Code samt ` Gravis`` und  
nur einem Doppelgravis: `` `` `.

## Fett formatiert

Ganz ähnlich wie kurze Codepassagen kann man Textstellen kursivieren und fetten, indem man sie mit einfachen beziehungsweise doppelten Sternchen oder Unterstrichen umgibt:

\*Kursive\* und **fette** Wörter,  
sowie fette und kursive.

Kursivierung und Fettung sind dabei lediglich die üblichen Darstellungsformen, die Auszeichnungen bedeuten eigentlich „betont“ und „stark betont“, was man sogar verschachteln darf:

Normal, kursiv und **fett-kursiv**\_.  
Auch **fett-kursiv**.

Etwas kompliziert werden die Regeln, wenn die Auszeichnungen nicht an Leerzeichen enden. Um einen Teil eines Wortes zu kursivieren, eignen sich zum Beispiel nur Sternchen (Wort\*teil\*kursivierung). Unterstriche funktionieren hier nicht, damit Markdown nicht häufige Konstruktionen wie Dateinamen mit Unterstrichen versehentlich auszeichnet. Wenn es zu einer falschen Auszeichnung kommt, hilft wieder der Backslash:

Ein Name\_mit\_Unterstrichen und  
ein Name\*mit*Sternchen.

CommonMark spezifiziert keine weiteren solchen Formatierungen. Allerdings gibt es wieder einige Ergänzungen, bei denen es sich lohnt, einfach auszuprobieren, ob ein System sie unterstützt: Mitunter kann man mit einfachen Tilden Text tief- (~1~) und mit Zirkumflexen hochstellen (^2^). Recht häufig markieren doppelte Tilden Text als gelöscht; er wird dann durchgestrichen angezeigt. Manche Systeme erlauben auch, Text mit doppelten Gleichheitszeichen hervorzuheben. Er bekommt dann etwa einen gelben Hintergrund:

Text mit einer ~~~~Löschung~~~~ und  
einer ==Hervorhebung==.

## Links und verlinkte Bilder

Links setzt man in Markdown mit einer Reihe von Syntaxvariationen, die aus Klammern bestehen. Eine Möglichkeit ist, den zu verlinkenden Text in eckige und das Linkziel in runde Klammern zu setzen:

Webseite des [Magazins für  
Computertechnik](https://ct.de).

Was man als Linkziel notiert, ist Markdown egal. Häufig handelt es sich um URLs oder Dateipfade – was in der Ausgabe funktioniert, hängt vom jeweiligen System ab. Zusätzlich zum Ziel kann man auch einen Link-Titel angeben. Den schreibt man einfach, von (mindestens) einem Leerzeichen getrennt, dahinter

und fasst ihn in Anführungszeichen oder runde Klammern ein:

```
[Linktext](Linkziel "Linktitel")
```

Bei einer Ausgabe als HTML-Code dient der Linktitel als Inhalt des `title`-Attributes des Links.

Linkziel und `-titel` direkt nach dem verlinkten Text anzugeben kann unübersichtlich werden, besonders wenn man sehr viele Links oder Links mit sehr langen URLs in einen Text einbettet. Abhilfe schaffen Referenzenlinks, die nach dem verlinkten Text nur ein kurzes Label angeben. Um es von einem Linkziel zu unterscheiden, notiert man das Label in eckigen statt runden Klammern.

Irgendwo anders im Markdown-Dokument (auch davor) definiert man eine passende Linkreferenz. Die wiederholt das Label (ebenfalls in Klammern) und gibt nach einem Doppelpunkt das Linkziel und den optionalen Titel an:

```
Webseite des [Magazins für  
Computertechnik][CT].
```

```
[CT]: https://ct.de "c't-Website"
```

Wenn das Label mit dem Linktext übereinstimmt, darf man es auch weglassen und die Referenz funktioniert trotzdem; die eckigen Klammern des Labels sind dann optional. Praktischerweise ignoriert Markdown Groß- und Kleinschreibung bei der Label-Zuordnung:

```
Webseiten der [ct][] und von [heise].
```

```
[CT]: https://ct.de
```

```
[Heise]: https://heise.de
```

Als letzte Linkvariante gibt es noch eine Syntax für den Fall, dass Linktext und `-ziel` identisch sind. Um dann nicht `[https://ct.de](https://ct.de)` schreiben zu müssen, kann man auch einfach das Linkziel in spitze Klammern setzen: `<https://ct.de>`. Einige Markdown-Systeme erfordern nicht

einmal das und verlinken automatisch alles, was sie als URL erkennen.

Eng verwandt mit Links sind Bilder – zumindest aus Markdown-Sicht: Eine Bilddatei kann man nicht sinnvoll in eine Textdatei einbetten und muss sie daher wie ein Linkziel referenzieren. In Markdown geschieht das über dieselbe Syntax, der man lediglich ein Ausrufezeichen voranstellt:

```
![Beschreibung](Pfad/zum/Bild-1.png)
```

Mit der Syntax von Referenzenlinks funktioniert das ebenso, nur die Variante mit spitzen Klammern gibt es nicht für Bilder. Anstelle des Linktexts steht eine Bildbeschreibung. Üblicherweise wird die als Alt-Text des Bildes ausgegeben, sollte also den Bildinhalt erklären. Allerdings weichen einige Markdown-Systeme davon ab und geben Bilder, die für sich alleine in einer Zeile stehen, anders aus. Die Beschreibung rendern diese Systeme dann als Bildunterschrift. Wieder probiert man am besten einfach aus, wie das jeweilige System diesen Fall handhabt.



Wer lange Texte in Markdown schreibt, sollte sich darauf spezialisierte Editoren ansehen. Die können – wie hier das

Programm „Apostrophe“ – zum Beispiel die layoutete Vorschau neben dem Markdown-Quelltext anzeigen.

## HTML einbetten?

Das deckt die wesentlichen Aspekte von Markdown und CommonMark ab – bis auf einen. Wie eingangs erwähnt, war Markdown ursprünglich (nur) dafür gedacht, Texte für das Web zu verfassen. Der Output einer Markdown-Implementierung sollte daher HTML sein und schon im Eingabetext vorhandenes HTML konnte die Software einfach durchreichen. Das ist im Grunde immer noch der Fall; wer HTML beherrscht, kann es einfach im Markdown-Text notieren:

```
<div class="test">
```

```
*Kursiver Text*
```

```
</div>
```

Ob das Ergebnis wie erwünscht ausfällt, ist allerdings eine andere Frage. Zwar produzieren die meisten Markdown-Anwendungen tatsächlich HTML (zumindest als Zwischenschritt), aber häufig filtern sie ihre Eingabe – etwa aus Sicherheitsgründen – und man kann keine (beliebigen) HTML-Tags in das Ergebnis schleusen.

Außerdem sind die genauen Regeln zum Mischen von Markdown- und HTML-Code komplex. Im obigen Beispiel sind etwa die Leerzeilen unabdingbar. Durch sie wird die mittlere Textzeile als Markdown und nicht als HTML aufgefasst, sodass die Sternchen Wirkung entfalten – zumindest in manchen Implementierungen. Weil solche Sprachmischungen schnell implementierungsabhängige Ergebnisse produzieren, die man kaum noch durchschaut, sollte man sie eher als Notlösung betrachten: Kann man ausprobieren, wenn man mit Markdown-Syntax alleine einfach nicht ans Ziel kommt.

# Tabellen

Als Beispiel für so einen Fall nennt die originale Markdown-Spezifikation Tabellen. Das ist zum Glück meistens nicht mehr korrekt, denn HTML-Tabellen sind lästig zu schreiben und schwer zu lesen. Viele Markdown-Implementierungen unterstützen eine simplere und viel übersichtlichere Syntax:

```
| Magazin | Zyklus   | Domain           |
| - - - - - | - - - - - | - - - - - - - - - |
| c't     | 14 Tage  | ct.de           |
| iX      | 4 Wochen | ix.de           |
| Mac & i | 8 Wochen | mac-and-i.de   |
```

Allerdings sind Tabellen kein Teil der CommonMark-Spezifikation und es gibt zahlreiche Syntaxvariationen. Meistens darf man zum Beispiel die äußeren senkrechten Striche (|) auch weglassen. In der Regel kann man auch bestimmen, wie Tabellenspalten ausgerichtet werden sollen, indem man Doppelpunkte in die Trennlinie zwischen der Kopfzeile und dem Rest der Tabelle setzt. Beides zusammen sieht so aus (die erste Spalte wird linksbündig, die zweite zentriert und die dritte rechtsbündig ausgegeben):

```
Magazin | Zyklus   | Domain
: - - - - - | : - - - - - : | - - - - - - - - - :
c't     | 14 Tage  | ct.de
iX      | 4 Wochen | ix.de
Mac & i | 8 Wochen | mac-and-i.de
```

Außerdem bietet Markdown wieder mal eine Abkürzung für Fauler: Man muss die Tabellenspalten nicht im Quelltext ausrichten, damit sie korrekt interpretiert werden. Das vereinfacht das Tippen erheblich, geht aber deutlich zulasten der Lesbarkeit:

```
Magazin | Zyklus | Domain
:- | :-: | -:
c't | 14 Tage | ct.de
iX | 4 Wochen | ix.de
Mac & i | 8 Wochen | mac-and-i.de
```

Neben Tabellen gibt es noch viele weitere Markdown-Ergänzungen, die sich weder in CommonMark noch der originalen Spezifikation der Sprache finden. Dazu gehören zum Beispiel Fußnoten oder Textabschnitte, die – häufig am Anfang des Dokuments – Metadaten enthalten. Manche Systeme erlauben auch Emojis und sogar mathematische Formeln in LaTeX-Syntax im Markdown-Text.

Wer Bedarf an solchen Erweiterungen verspürt, konsultiert am besten die Dokumentation der Software, um die es geht. Im Regelfall ist das der schnellste Weg, um herauszufinden, was im konkreten Fall möglich ist und welche Syntaxvariante das System verlangt. Um einfach nur einen Blick über den Tellerrand zu werfen, eignet sich die Dokumentation von Pandoc (siehe [ct.de/y5hr](https://ct.de/y5hr)), einem System, das zahlreiche Ergänzungen in zahlreichen Syntaxvarianten versteht [7].

## Fazit

Wer die Markdown-Basics beherrscht, dem steht in vielen verschiedenen Systemen eine einheitliche, praktische und schnelle Eingabemethode zur Verfügung. Zwar unterscheiden sich Implementierungen in Details, aber wenn gelegentlich etwas nicht wie erwartet funktioniert, dann hilft in der Regel ein Backslash, eine zusätzliche Leerzeile, um Elemente zu trennen, oder man vereinfacht die Dokumentstruktur etwas: Codeblöcke in Listen in Zitaten in Listen bringen nicht nur so manche Markdown-Implementierung in die Bredouille – sie sind auch einfach schwer zu verstehen.

Wie zahlreiche andere c't-Artikel entstand auch dieser in einem Markdown-Editor. Das ging wie immer flott und einfach von der Hand und ist auch im Quelltext exzellent zu lesen – obwohl es wirklich verschärfte Bedingungen sind, mit Markdown-Syntax über Markdown-Syntax zu schreiben. ([syt@ct.de](mailto:syt@ct.de))

1. Literatur
2. [Stefan Wischner, Blitznotizen, c't 13/2021, S. 82](#)

3. [Achim Barczok, Notiznetz, c't 6/2021, S. 84](#)
4. [Liane M. Dubowy, Organisationstalent, Schreiben und organisieren in Notion, c't 12/2021, S. 166](#)
5. [Andrijan Möcker, Tippgemeinschaft, HedgeDoc: Gemeinsam texten mit Markdown-Pads, c't 13/2022, S. 164](#)
6. [Jan Mahn, Nach Art des Hauses, Rezeptdatenbank selbst hosten mit Tandoor Recipes, c't 2/2022, S. 164](#)
7. [Anna Simon, Website-Wasservogel, Mit Pelican einfach schnelle Webseiten generieren, c't 12/2022, S. 164](#)
8. [Jan Mahn, Formatautomat, Automatische Textumwandlung mit pandoc, c't 7/2018, S. 168](#)

**Markdown-Spezifikationen:** [ct.de/y5hr](https://ct.de/y5hr)

---

# Neuer Ärger für Facebook

## Neuer Ärger für Facebook



Laut der irischen Datenschutzbehörde muss Facebook den Transfer persönlicher Daten von EU-Bürgern in die USA stoppen.

*Bild: dpa*

Die irische Datenschutzbehörde DPC will Facebook **untersagen, Nutzerdaten aus der EU in die USA zu übertragen**. Darüber hat die DPC, in deren Jurisdiktion Meta fällt, den Europäischen Datenschutzausschuss informiert. Da die übrigen EU-Datenschutzbehörden noch Einspruch erheben können, erwarten Experten jedoch einen Aufschub. Der Gründer der Datenschutzorganisation noyb, Max Schrems, geht von mindestens einem Jahr aus; auch, weil die Konzernmutter Meta voraussichtlich klagen wird. Die DPC gründet ihre Entscheidung auf das EuGH-Urteil „Schrems II“, in dem die Richter das „Privacy Shield“-Abkommen für rechtswidrig erklärten. Die seither verwendeten Standardvertragsklauseln reichen laut DPC nicht aus.

Kritik musste Meta auch nach der Veröffentlichung seines Menschenrechtsberichts für 2020 und 2021 einstecken: So betrachte Meta Handlungsempfehlungen der federführenden Kanzlei nicht als verbindlich; zugleich soll die Lage in Indien laut der NGO India Civil Watch International geschönt worden sein – interne Dokumente hätten gezeigt, dass Meta nicht angemessen auf schädliche Inhalte reagierte. ([mon@ct.de](mailto:mon@ct.de))