

Einleitung

Ich arbeite schon lange mit dem Internet. Wenn ich schreibe *lange*, dann meine ich *lange* – mein erster Browser lief über die Kommandozeile. So wie meine erste Backend-Entwicklung ein CGI-Skript war. In den fast drei Jahrzehnten, in denen ich Dutzende von Projekten für das Web entwickelt habe, habe ich eines festgestellt: Am wertvollsten ist die Fähigkeit, die Lücken zwischen den Welten des Designs, der User Experience und der Entwicklung zu schließen.

Jeder dieser Bereiche zieht Fachleute mit unterschiedlichem Bildungs- und/oder Hobbyhintergrund an. Dies kann zu Spannungen führen, da die Leute in jedem Lager dazu neigen, zu denken, dass ihr eigenes Gebiet den größten Einfluss auf den Erfolg eines Webprojekts hat. Die Realität ist, dass das Web im Zusammenspiel dieser Bereiche existiert und sich auf alle Bereiche gleichermaßen stützt. Ich habe die Erfahrung gemacht, dass die wertvollsten Mitarbeitenden in einem Webprojekt diejenigen sind, die genug von jedem Bereich verstehen, um in diesem Spannungsfeld zu agieren und sicherzustellen, dass jedes Team effektiv mit anderen Teams kommuniziert und alle gemeinsam auf ein Ziel hinarbeiten.

Das bringt mich zu Stephanie Stimacs »Design trifft Code«. Dieses Buch ist eine unschätzbare Ressource für Entwickler, die ihr Verständnis dafür verbessern wollen, was es braucht, um heute für das Web zu entwickeln. Es bietet einen Crash-Kurs in Design und User Experience auf eine Art und Weise, die sowohl Ihr Wissen über diese Themen verbessert als Ihnen auch die notwendigen Fähigkeiten vermittelt, um in diesen Bereichen produktiv zu werden. Über diese oberflächliche Perspektive hinaus bietet dieses Buch jedoch die Grundlage, Ihre Karriere voranzutreiben und zu einer Person zu werden, die im Spannungsfeld zwischen Design, User Experience und Entwicklung wächst. Stephanie Stimac ist die perfekte Person, um Ihnen diese Erkenntnisse zu vermitteln. Sie ist nicht nur eine exzellente Kommunikatorin, sondern verfügt auch über jahrelange Berufserfahrung als Designerin, Entwicklerin, User-Experience-Expertin und Dozentin. Sie hat sogar hinter den Kulissen an Webbrowsern und den darin implementierten Webstandards gearbeitet. Das vermittelt ihr ein unglaublich tiefes Verständnis für das Web und die Tools, auf die wir uns bei dessen Aufbau verlassen. Ganz gleich, ob Sie ein langjähriger Entwickler sind, der einen neuen Zweig in Ihrem Baum der Fähigkeiten wachsen lassen

möchte, oder ob Sie neu anfangen und sich eine solide Grundlage für die Entwicklung von Webanwendungen aneignen möchten, »Design trifft Code« wird Ihnen mit Sicherheit eine Fülle von Wissen vermitteln, das Ihnen auf Ihrem Weg hilft.

Aaron Gustafson

Autor von »Adaptive Web Design:

Crafting Rich Experiences with Progressive Enhancement«

Vorwort

Ich habe schon lange vor meinem Studium angefangen, für das Internet zu gestalten. Ich habe meine Vorliebe für das Anpassen von Dingen wie Myspace und LiveJournal entdeckt und begonnen herauszufinden, wie man Photoshop benutzt und meine Webseiten mit CSS bearbeitet. Als Teenager war das nur ein Hobby. Erst in meinem letzten Jahr an der Highschool habe ich Grafikdesign als Berufswunsch in Betracht gezogen. Eigentlich wollte ich Modedesignerin werden, aber mein Portfolio war eher für Grafikdesign geeignet.

An der Universität belegte ich im Studiengang Design den Schwerpunkt *New Media Design*, der Web, Interaktivität, Video und Spezialeffekte umfasste. Ich erinnere mich zwar daran, dass ich die Grundlagen von HTML und CSS gelernt habe, aber das Tool, mit dem ich viel Zeit verbracht habe, war Adobe Flash und die Sprache ActionScript 3.0. Perfektes Timing: Als ich 2010 meinen Abschluss gemacht hatte, war Flash als Medium für das Web kurz vor dem Aussterben. Meine ersten beiden Design-Jobs waren kurz. Dann wechselte ich in eine Rolle, die der Auslöser für meinen Einstieg in die Frontend-Entwicklung war. Ich beschäftigte mich mit User-Experience-Forschung, User-Experience-Design, visuellem Design und Frontend-Entwicklung und arbeitete mit einem Entwickler für komplexere Backend-Funktionen und JavaScript zusammen. In meiner Rolle als Frontend-Entwicklerin und in der Zusammenarbeit mit Vollzeit-Entwicklern stieß ich auf das Problem, dass Designer großartige Ideen hatten, die aber zu dem Zeitpunkt nicht umzusetzen waren, und sie deshalb das Design änderten.

Ich war vier Jahre lang in dieser Funktion tätig, bevor ich zum Microsoft-Edge-Team wechselte. Dort machte ich denselben Job, jedoch mit einer anderen Art von Freiheit. Ich hatte mehr Raum zum Lernen und mehr Zeit, die ich mit den Entwicklern verbringen konnte, um ihre Erfahrungen und die Probleme zu verstehen, die sie hatten. Ich habe gelernt, wie man Git verwendet, und Änderungen einpflegt (und habe oft Hilfe gebraucht, um Dinge mit Node und npm zurückzusetzen. Danke, Anton, dass du das für mich wieder in Ordnung gebracht hast). Ich habe Tools für Entwickler entworfen, Developer Experiences gebaut und begonnen, mich mit Standards und Funktionen von Webplattformen zu beschäftigen. Das führte zu meiner Tätigkeit als Produktmanagerin für Developer Relations.

Ich finde mich oft an einer Kreuzung wieder, und zwar an einer sehr belebten – eine Designerin, die eine Designerin wurde, die programmieren kann, die eine Produktmanagerin wurde, die Tools für Entwickler entwarf und entwickelte, die eine Produktmanagerin wurde, die Webplattform- und Browserfunktionen propagierte, während ich weiterhin die Teile entwarf und entwickelte, die ich für diese Arbeit überhaupt brauchte. Letztendlich sehe ich mich aber als Bindeglied zwischen Entwicklern und Designern. Ich verstehe beide und kann mich in beide hineinversetzen.

In meiner Zeit als Produktmanagerin für Developer Relations habe ich festgestellt, dass sich einige Webentwickler wünschten, sie könnten die von ihnen erstellten Webseiten besser gestalten. Häufig habe ich gehört, dass sie zwar eine Webseite erstellen konnten, diese aber nie richtig aussah, und dass die Arbeit mit Farben schwierig war. Ich habe diese Dinge beiseitegeschoben; sie waren zu diesem Zeitpunkt keine lösbareren Probleme. Aber eines Tages hatte ich eine E-Mail von Manning in meinem Posteingang, und plötzlich waren diese Designprobleme von Webentwicklern meine Probleme, die es zu lösen galt.

Danksagung

Ich habe mit dem Schreiben dieses Buches Ende 2020 begonnen, als es so aussah, als ob ich viel Zeit zum Schreiben haben würde. Und das hatte ich auch. Aber das Leben kommt einem oft in die Quere, und zeitweise war es schwierig, in den letzten zwei Jahren so viele Veränderungen im Leben und ein Buch unter einen Hut zu bringen. Ich möchte meinen Freunden danken – ihr wisst, wer ihr seid –, die mich in vielen Zoom-Anrufen, in unserer Slack-Gruppe und persönlich ermutigt haben. Ihr seid meine Seelenfamilie und erhellt mein Leben ständig mit eurer Kreativität.

Danke an meine Familie, die mich bei diesem Projekt unterstützt hat und immer wieder nachgefragt hat, wie es mir geht: Ich liebe euch. Und obwohl sie dies nie lesen kann, danke ich meiner Hündin Vogue, die viele Nächte auf meinem Schoß saß, während ich geschrieben habe, und mich in Schach hielt, wenn ich zu lange am Computer saß.

Als Nächstes möchte ich meiner Lektorin bei Manning, Sarah Miller, danken. Ich danke Ihnen für Ihre Geduld und Ihr Verständnis, als ich beim Schreiben dieses Buches mehrere Hürden überwinden musste. Wir haben im Laufe des Buches mehrere Herausforderungen gemeistert, da es sich nicht um ein typisches Fachbuch handelt. Danke, dass Sie so offen waren, wenn ich anderer Meinung war, und für all Ihre Ratschläge während dieses Prozesses. Ich möchte auch den übrigen Mitarbeitenden von Manning danken: meinem Redakteur für die technische Entwicklung, Kris Athi, meinem Redakteur für die Rezensionen, Aleks Dragosavljevic, meiner Produktionsredakteurin, Deirdre Hiam, meiner Lektorin, Alisa Larson, und meiner Korrektoren, Melody Dolab.

Vielen Dank an alle Rezessenten und diejenigen, die mit ihrem Feedback dazu beigetragen haben, dass das Buch einen besseren roten Faden bekommt: Adrian Rossi, Al Krinker, Alain Couniot, Alex Lucas, Andres Sacco, Arun Kumar, Bang Nteme, Bruno Sonnino, Cass Petrus, Daniel Couper, Daniel Carl, Danilo Zekovic, David Cabrero Souto, David Krief, Frans Oilinki, Harald Kuhn, Harrison Maseko, Håvard Wall, Jereme Allen, Jose San Leandro, Katia Patkin, Marcin Sek, Martin Tidman, Miguel Isidoro, Nick Rakochy, Oliver Korten, Phillip Sorensen, Ramaa Kandasamy, Rodney Weis, Sezin Çagil, Simeon Leyzerzon, Søren Dines Jensen, Steve Albers, Steve Prior, Tanuj Shroff, Tanya Wilke, Viorel Moisei und Weyert de Boer.

Und schließlich danke an meine Online-Community und diejenigen, die ich persönlich kenne. Ohne eure Ermutigung hätte ich nie angefangen, auf Konferenzen zu sprechen, und ich hätte sicher nicht einmal daran gedacht, ein Buch zu schreiben. Ich bin gesegnet, dass ich so viele wunderbare Menschen kenne, die ich in meinem Beruf als Freunde betrachte. Ich danke euch für all eure Unterstützung und Ermutigung.

Über dieses Buch

»Design trifft Code« wurde geschrieben, um Designgrundlagen, Grundlagen der User Experience und die Schlüsselemente des Designs zu vermitteln, die eine Webseite zum Leben erwecken und ihr Charakter verleihen. Es lehrt, wie Designelemente verwendet werden können, um einen Ton und eine Botschaft zu vermitteln, und wie sich dieser Ton und diese Botschaft durch Designelemente verändern lassen. Dieses Buch ist in der Abfolge eines typischen Projektzyklus aufgebaut. Es beginnt mit den grundlegenden Designprinzipien, die benötigt werden, bevor die Phase der User Experience oder des visuellen Designs beginnt. Es geht in die Phase der User Experience über, die sowohl die Forschung als auch das Design umfasst, mit den Grundlagen für die Anwendung visueller Designelemente in den folgenden Kapiteln. Den Abschluss bilden das Testen des Designs, der iterative Zyklus und die Entwicklung, das Kernstück des Designprozesses, das den Entwurf ins Web bringt und nutzbar macht. Das ist das Herzstück der gesamten User Experience einer Website.

Wer sollte dieses Buch lesen?

»Design trifft Code« ist für Webentwickler geschrieben, die Webseiten sowie Webanwendungen erstellen und die Grundlagen darüber lernen möchten, was eine Webseite oder Webanwendung gut aussehen lässt und wie man Designentscheidungen mit Selbstvertrauen trifft. Ganz gleich, ob Sie freiberuflicher Entwickler sind oder in einem Team mit Designern arbeiten, der Inhalt dieses Buches wird Ihnen die Werkzeuge und das Vokabular an die Hand geben, um mit Ihren Kunden oder Ihrem Designteam über Design zu sprechen und darüber, wie es mit Ihrer Arbeit als Entwickler zusammenhängt – ganz gleich, ob Sie Anfängerin oder Fortgeschrittenen sind. Dieses Buch konzentriert sich darauf, Entwicklern die Grundlagen des visuellen Designs und der User Experience im Zusammenhang mit dem Web zu vermitteln und zu zeigen, wie sie diese Grundlagen auf ihre aktuelle oder zukünftige Rolle anwenden können. Der Begriff Webentwickler kann in der heutigen Berufswelt viele verschiedene Bedeutungen haben. Entwickler, die am meisten von diesem Buch profitieren werden, sind Entwickler, die Code schreiben, der letztendlich das Frontend einer Website oder das, was der Benutzer

sieht, beeinflusst. Diese Entwickler haben Berufsbezeichnungen wie Full-Stack-Entwickler, Frontend-Entwickler oder Web-App-Entwickler.

Wir werden uns nicht stark auf Code konzentrieren. Aber da es in diesem Buch um Design und User Experience für das Web geht, wird es einige Code-Referenzen geben. Die meisten Menschen, die von diesem Buch profitieren werden, haben einen starken Code-Hintergrund in JavaScript oder PHP und haben ein Verständnis von HTML und CSS mit wenig oder gar keinem Wissen über Design oder User-Experience-Grundlagen.

Wie dieses Buch aufgebaut ist: Ein Fahrplan

Teil 1 befasst sich mit den Vorteilen, Designprinzipien zu verstehen, und mit den wichtigsten Designprinzipien selbst. In Kapitel 1 wird erörtert, warum Entwickler Designgrundlagen lernen sollten und welche positiven Auswirkungen dies auf die Kommunikation zwischen Teams haben kann. Kapitel 2 befasst sich eingehend mit den wichtigsten Designprinzipien. Sie sind grundlegend für das Verständnis dessen, was ein Design ausgefeilt und gut organisiert erscheinen lässt.

Teil 2 beginnt den Designzyklus mit den Grundlagen der User Experience. Kapitel 3 ist ein einführender Überblick über die vielen Facetten der User Experience, einschließlich Forschung, Werbetexten und Design. Kapitel 4 befasst sich mit der Forschungsphase und gibt einen Überblick darüber, warum Nutzerforschung (User Research) notwendig ist, welche verschiedenen Arten von Daten gesammelt werden können und wie einige der gängigsten Übungen zur Datenerfassung aussehen. In Kapitel 5 beginnen wir damit, die Grundlage für unser Design zu legen, indem wir Inhalte organisieren und Nutzerströme entwerfen, wobei wir uns auf die Phase des User-Experience-Designs konzentrieren.

Teil 3 stellt weitere Möglichkeiten vor, wie man über das Layout nachdenkt und wie man die Kernelemente des Designs aufbaut, um die Grundlagen aus Kapitel 5 zum Leben zu erwecken. Kapitel 6 behandelt gängige Layouttypen, das Einrichten eines Rasters, Lesemuster und Layout-Überlegungen bei der Gestaltung der responsiven Version Ihrer Website oder App. Kapitel 7 befasst sich mit Animationen und zeigt, wie Sie Ihr Layout und Ihre Benutzeroberfläche mit Animationen aufwerten können und warum diese nicht nur ein nachträglicher Gedanke sein sollten. In Kapitel 8 geht es um die Grundlagen der Typografie und darum, wie Sie mit der Wahl der Schriftart den Ton Ihrer Website ändern können. In Kapitel 9 geht es vor allem um Farben und die Auswahl von Bildmaterial in Abhängigkeit von Ihrer Farbpalette. In Kapitel 10 fassen wir alles zusammen und nutzen das in den Kapiteln 4 bis 9 Gelernte, um eine Homepage mit einem mehrschichtigen Ansatz zu erstellen und zu gestalten, der den Kapiteln dieses Buches folgt.

Teil 4 schließt den Designzyklus ab, in dem es um das Testen von Designs geht und darum, warum die Entwicklung das Rückgrat der Benutzererfahrung im Web ist. In Kapitel 11 geht es um das Testen Ihrer Entwürfe und die Bestätigung, dass Sie die Ziele Ihrer Website erreichen. Kapitel 12 beendet das Buch mit Verbindungen und Überlegungen zu User Experience und Entwicklung.

Teil 1

Grundlagen der Gestaltung

Im ersten Teil des Buches lege ich den Grundstein für den Rest des Buches und stelle Gestaltungsprinzipien vor, die Sie unbedingt verstehen müssen, bevor Sie mit der visuellen Gestaltungsphase eines Projekts beginnen.

In Kapitel 1 erkläre ich den Zweck dieses Buches und warum es für Sie als Entwickler von Vorteil ist, die Designgrundlagen zu verstehen. Kapitel 2 bildet die Grundlage des Buches und die Gestaltungsprinzipien, die wir erforschen werden. Jedes Prinzip, das in diesem Kapitel erläutert wird, ist wichtig, um zu verstehen, wie man Beziehungen zwischen Inhalten herstellt, die in Teil 2 vorgestellt werden. Diese Prinzipien können auf alle visuellen Elemente angewendet werden, was wir in Teil 3 tun werden. Je besser Sie sich mit den Gestaltungsprinzipien vertraut machen, desto radikaler können Sie diese Prinzipien anwenden, um herausragende Designs zu erstellen. Am Ende dieses Teils werden Sie in der Lage sein, die User-Experience-Phase des Designzyklus zu beginnen.

1 Die Kluft zwischen Design und Entwicklung überbrücken

Dieses Kapitel behandelt

- Die Gesamtheit des in diesem Buch behandelten Webdesign-Prozesses
- Die sich wandelnde Rolle der Webentwickler
- Die Vorteile für Freiberufler und fest angestellte Entwickler, wenn sie Designgrundlagen und User-Experience-Prozesse verstehen

Das Web ist ein Ort des Wandels. Jedes Jahr werden neue Technologien und Frameworks eingeführt. HTML und Cascading Style Sheets (CSS) bilden nach wie vor die Grundlage für das Erstellen von Webseiten und Webanwendungen. Beides wird ständig weiterentwickelt, um den Anforderungen der Entwickler gerecht zu werden, wenn sich Design weiterentwickelt.

Ein Teil der Webentwicklung besteht darin, mit den dafür verwendeten Technologien auf dem Laufenden zu bleiben. Aber wie funktioniert das beim Design? Obwohl sich die Werkzeuge im Laufe der Zeit ändern, sind viele der zugrunde liegenden Konzepte zeitlos. Was Sie sehen, ist eine Verschiebung der Trends im Laufe der Zeit. Meine Kenntnisse über Webentwicklung sind für mich als Spezialistin für Webdesign von unschätzbarem Wert. Auf die Frage, ob Designer programmieren lernen sollten, lautet meine Antwort: Ja. Das Verständnis dafür, was im Web möglich ist, ist von großem Vorteil. Es macht die Kommunikation zwischen Designern und Entwicklern viel einfacher. In der Technik gilt die Kommunikationslücke zwischen Designern und Entwicklern als eine der am schwersten zu überbrückenden. Eine weniger häufig gestellte Frage lautet: »Sollten Entwickler lernen zu entwerfen?« Auch hier lautet meine Antwort: Ja. Zumindest wenn es um das Verständnis von Designgrundlagen geht. Wenn Entwickler über Designkenntnisse verfügen, können sie eine Benutzeroberfläche schnell aktualisieren, während sich ein Designteam mehr auf die Lösung von User-Experience-Abläufen konzentriert als auf kleine Anpassungen an Pixeln. Ein CEO teilte meinem Entwicklungsteam einmal mit,

dass sich die Benutzeroberfläche eines unserer Tools nicht so effizient anfühlte wie die einiger unserer Mitbewerber. Er meinte damit, dass es zu viel Leerraum zwischen den Komponenten gab. Die Benutzeroberfläche war nicht so raffiniert, wie sie sein könnte. Einer der Entwickler reduzierte die Abstände zwischen den Komponenten und straffte das Design. Er brauchte keinen Designer, um Mock-ups zu erstellen. Stattdessen konnte er eine Änderung vornehmen, sie einem Designer zeigen, um sicherzustellen, dass die Teams sich bei der Änderung einig waren, und dann die Änderungen an der Benutzeroberfläche umsetzen.

Wenn Entwickler Ihr Produkt mit einem anderen Design vergleichen, den Unterschied – in diesem Fall die Verwendung von Leerraum – erkennen und eine Änderung vornehmen können, können Sie sich als Designer auf die Lösung komplexerer Probleme konzentrieren. Ein guter Entwickler weiß, dass sich kleine visuelle Änderungen positiv auf das Erstellen eines ausgefeilten Designs auswirken.

1.1 Wie Entwickler von Design- und User-Experience-Grundlagen profitieren

Designer und Entwickler sprechen in der Regel unterschiedliche Sprachen und haben unterschiedliche Fähigkeiten. Designer konzentrieren sich auf den Benutzer und die Gestaltung des Bildmaterials. Entwickler konzentrieren sich auf die Umsetzung von Bildern in etwas Interaktives für Kunden. Die beiden Bereiche überschneiden sich jedoch, da Entwickler das, was ein Designer entworfen hat, mit Code nachbilden, in der Regel mit CSS. Das Verständnis für das *Was* und *Warum* hinter dem, was man zum Leben erwecken möchte, macht einen zu einem besseren Entwickler und gibt das nötige Selbstvertrauen, um Designentscheidungen zu treffen, wenn man keinen Designkollegen hat. Design ist viel mehr als nur Optik und Ästhetik. Diese sind zwar wichtig und das Erste, was Kunden bemerken, wenn sie auf Ihrer Webseite landen. Aber der Rest des Erlebnisses ist genauso wichtig. Wenn die Webseite nur langsam lädt oder die Besucher die von Ihnen erwarteten Hauptaufgaben, wie den Kauf eines Artikels, nicht ausführen können, reicht die Ästhetik nicht aus. Sie müssen wissen, wie Sie Entscheidungen über die User Experience treffen können, die wiederum die Entscheidungen über das visuelle Design beeinflussen. Als ich vor über zehn Jahren meine Karriere begann, sah der typische Arbeitsablauf für ein Websiteprojekt folgendermaßen aus:

1. Der Designer verglich für das Projekt die Websites der Konkurrenten seines Kunden und beurteilte, was gut gemacht war und was nicht.
2. Der Designer erstellte einige Benutzer-Personas der Personen, auf die das Projekt abzielt, für die Lösung eines bestimmten Bedürfnisses (z.B. Informationen liefern, ein Produkt kaufen, einen Termin buchen, Hotels vergleichen usw.).
3. Auf der Grundlage der Wettbewerbsforschung, der Benutzer-Personas und der geschäftlichen Anforderungen erstellte der Designer Wireframes, einfache Dia-

gramme der Website, die die Platzierung der Inhalte zeigen, und holte dann die Zustimmung des Kunden zu diesen Wireframes ein.

4. Der Designer ging schließlich in die visuelle Entwurfsphase des Projekts über, stellte dem Kunden pixelgenaue Mock-ups zur Überprüfung zur Verfügung, überarbeitete die Mock-ups, holte die Genehmigung ein und übergab das Projekt dann an den Entwickler, um die Website zu programmieren.
5. Der Entwickler nahm die statischen Mock-ups und erstellte die interaktive Website.
6. Nach einem Qualitätstest, bei dem auf Fehler oder Probleme geprüft wurde, wurde die Website veröffentlicht.

All diese Schritte finden auch heute noch statt. Dieser Arbeitsablauf ist weitgehend Gegenstand dieses Buches: der Projektzyklus vom Beginn der User-Experience-Phase über das User-Experience-Design (Wireframes) bis zur Anwendung visueller Designelemente, um das Design zum Leben zu erwecken. Wir werden diesen Projektzyklus jedoch anpassen, um uns auf moderne Arbeitsabläufe zu konzentrieren, die ein iteratives Design und eine iterative Entwicklung, engere Arbeitsbeziehungen und ständiges Testen und Forschen zur Verbesserung eines Designs fördern.

Der Arbeitsablauf in dem zuvor Aufgeführten ist zwischen dem Designer und dem Entwickler aufgeteilt. Ein User-Experience-Designer kann sich ausschließlich auf die Wettbewerbsforschung und das Wireframing konzentrieren, während er mit einem visuellen oder Interaktionsdesigner zusammenarbeitet, der sich auf Designentscheidungen wie Typografie, Farbe, Abstände und Animationen konzentriert – all die visuellen Aspekte einer Webseite, die ich in diesem Buch untersuchen werde. Nichtsdestotrotz wird die Arbeit zwischen Designern und Entwicklern aufgeteilt, wobei die Entwickler erst bei der Übergabe einbezogen werden.

Diese Arbeitsweise lässt keinen kollaborativen Ansatz zu. Entwickler verfügen über Einblicke und Kenntnisse, die Designer in der Regel nicht haben. Entwickler frühzeitig in den Prozess einzubeziehen, ist für den Projektzyklus von Vorteil. Egal ob es darum geht, unnötige Arbeit zu vermeiden, wenn ein Designmerkmal nicht implementiert werden kann, oder um eine andere Perspektive, die die kreative Richtung beeinflussen kann – es ist ein Gewinn für beide Seiten.

1.1.1 Zusammenarbeit und Kommunikation verbessern

Designer erwarten nicht, dass ihre Kollegen aus der Entwicklungsabteilung Designexperten werden, so wie Entwickler nicht erwarten, dass Designer Experten für JavaScript werden. Jeder von uns hat seine Rolle aus einem bestimmten Grund. Aber die Arbeit, die wir tun, ist miteinander verwoben. Der von den Entwicklern geschriebene Code kann sich auf das Design auswirken, und was die Designer entwerfen, kann sich auf die Entscheidungen auswirken, die die Entwickler beim Code treffen müssen. Manchmal gibt es auch eine Grenze für das, was von einem

Entwickler gebaut werden kann, entweder aufgrund technischer Beschränkungen oder einer Codelösung, die sich negativ auf das Benutzererlebnis auswirken würde, z.B. eine Animation, die das Laden der Webseite verlangsamt. Wenn man versteht, welche Entscheidungen für das Design ausschlaggebend sind, kann man die Zusammenarbeit und Kommunikation zwischen Designern und Entwicklern effizienter gestalten und das Hin und Her aufgrund dieser technischen Einschränkungen verringern. Wenn Entwickler mit den entsprechenden Werkzeugen und dem entsprechenden Vokabular ausgestattet sind, können sie den Designern mehr umsetzbares Feedback geben und die richtigen Fragen stellen. Sie können beginnen, das *Warum* hinter Designentscheidungen zu verstehen.

Dieses Buch soll dazu beitragen, Entwicklern die Terminologie und das Verständnis zu vermitteln, um während des gesamten Projektverlaufs effektiver mit einem Designteam zu kommunizieren. Auch wenn Sie nicht zu einem Unternehmen mit einem Designteam gehören, wird Ihnen die Lektüre dieses Buches das nötige Selbstvertrauen geben, um Designentscheidungen zu treffen, die zu einem gut aussehenden Endprodukt führen. Vielleicht sind Sie ein freiberuflicher Entwickler, der in der Lage sein möchte, visuelle Anpassungen an einer Webseite vorzunehmen, und Sie haben nicht das Budget, einen Designer zu engagieren (oder wollen es einfach nicht). Es kommt darauf an, diese wenigen Designprinzipien zu verstehen und sie anzuwenden.

1.1.2 Die Gründe für Designentscheidungen verstehen

Ich habe ein Projekt geleitet, das sich an Entwickler richtete. Es handelte sich um ein Tool, das den Code einer Webseite oder einer Webanwendung analysierte und Vorschläge zur Verbesserung der Accessibility, der Leistung und der Sicherheit einer Webseite gemacht hat. Obwohl es sich um ein Open-Source-Projekt handelte, waren die meisten Mitglieder des Teams, das daran arbeitete, Microsoft-Mitarbeitende. Ich wusste, dass Webentwickler Microsoft im Allgemeinen nicht immer positiv gegenüberstehen. Wenn also Microsoft ein weiteres Tool für Entwickler herausbringt, würden sie jedes Detail davon kritisch hinterfragen. Ich hatte beschlossen, dass unser Branding, ähnlich wie bei anderen Entwickler-Tools, ein illustriertes Maskottchen enthalten sollte, das wir in den Grafiken auf der Webseite verwenden konnten.

Mit dem Design des Maskottchens wollte ich bewusst ein Gefühl der Empathie bei den Nutzern unserer Webseite hervorrufen. Die Platzierung einer bestimmten Illustration im Benutzerfluss der Webseite, wie in Abbildung 1-1 dargestellt, war entscheidend, um dieses Gefühl zu erreichen. In den ersten Iterationen der Webseite wurden die Benutzer in eine Warteschlange gesetzt, um ihren Code analysieren zu lassen, da unser Backend nur eine bestimmte Anzahl an URLs auf einmal verarbeiten konnte. Menschen sind ungeduldige Wesen, wenn es um das Internet geht, und Entwickler noch viel mehr.

Ich ging davon aus, dass es Beschwerden über die Warteschlange geben würde, und so zeigte die Illustration auf dieser Seite unser Maskottchen in einem

dunklen Raum hinter einem Schreibtisch mit einer Lampe, mit schweißnassem Gesicht, während es sich durch die Warteschlange der URLs arbeitete.

Meine Hypothese wurde später bestätigt, als ich mit einem Entwickler über das Tool und die Website sprach und er mir sagte: »Ich wollte wütend sein, dass ich warten musste, aber ich hatte einfach Mitleid mit deinem Maskottchen.«

Einfühlungsvermögen für die Zielgruppe entwickeln, um bessere Erfahrungen zu ermöglichen

Weil ich die Zielgruppe unserer Website verstanden habe und mich in sie hineinversetzen konnte, war ich in der Lage, Designentscheidungen zu treffen, um künftige Beschwerden zu entschärfen. Schließlich wurde das Backend aufgerüstet, um mehr Codeanalysen gleichzeitig zu verarbeiten. Dies ist ein perfektes Beispiel dafür, wie Entwicklerentscheidungen und User Experience miteinander verwoben sind. In diesem Fall führte die anfängliche Einrichtung des Backends zu einer schlechten Erfahrung für die Nutzer der Website, da sie warten mussten. Bei den heute verfügbaren blitzschnellen Datenübertragungsgeschwindigkeiten sind die Menschen weniger geneigt, auf das Laden einer Webseite zu warten, bevor sie zur nächsten Option weitergehen. Die richtige Gestaltung dieser Erfahrung ist für die Geschäftskennzahlen entscheidend.

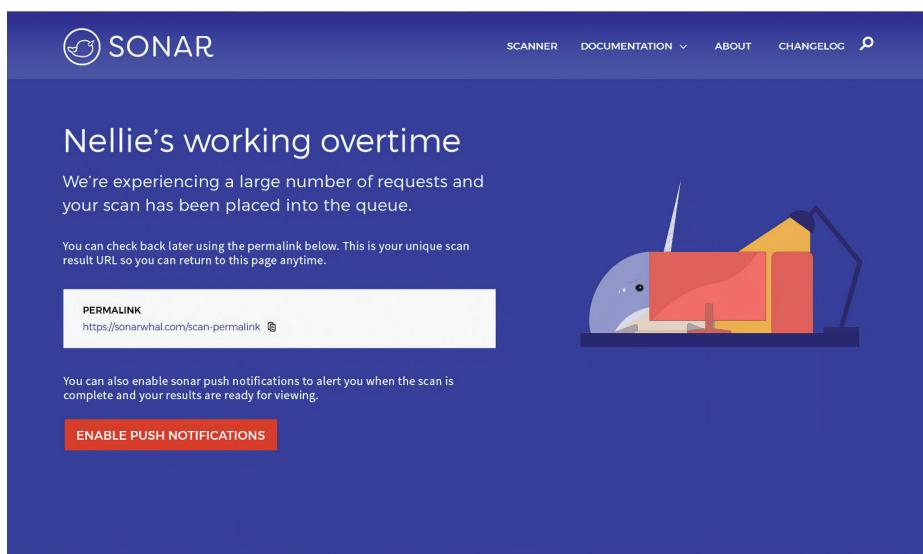


Abb. 1-1 Die Warteschlange für den Webseiten-Scanner zeigte das Narwal-Maskottchen, das Überstunden machte, um einen Webseiten-Scan für einen Entwickler zu erledigen.

Bei der Verbesserung der User Experience sind das Verständnis der Zielgruppe und die Fähigkeit des Entwicklers, sich in sie hineinzuversetzen, von unschätzbarem Wert. Im Fall der Codeanalyse-Webseite gab es mehrere Lösungen, die ein Entwickler ohne den Input eines Designers hätte umsetzen können. Sie hätten eine Ladeanimation oder eine Zwischenseite mit dem Hinweis einbauen können wie: »Schauen Sie später wieder vorbei, um Ihre Ergebnisse zu sehen«. Dies sind zwar nicht unbedingt schlechte Erfahrungen. Aber es sind Standarderfahrungen, die die Zielgruppe nicht berücksichtigen, die wenig Toleranz für die Verwendung eines Tools haben, insbesondere eines neuen Tools, das bei der ersten Verwendung einen guten Eindruck hinterlassen muss.

Technische Entscheidungen sind Entscheidungen für die User Experience

Wenn man die Grundlagen der Gründe für das Design und die User Experience versteht, kann man als Entwickler besser verstehen, wie sich Entscheidungen – vom Code-Stack über die Kompilierung des Codes bis hin zum Hosting-Anbieter – letztendlich auf die User Experience einer Webseite oder Webanwendung auswirken. Die technischen Entscheidungen sind ebenso Teil des Designs einer Webseite wie die visuelle Gestaltung. Deshalb ist es so wichtig, über die Designentscheidungen eines Designers aus der Sicht der Entwicklung zu sprechen.

Nehmen wir an, ein Designer entwirft eine Webseite, die viele Animationen mit Benutzerinteraktionen enthält. Die Sichtweise des Entwicklers, wie sich dies auf die Leistung der Webseite auswirkt, ist ein wichtiges Feedback, das dem Designer bereits zu Beginn des Prozesses gegeben werden sollte. Sind alle Animationen ein notwendiger Bestandteil der User Experience der Webseite? Wenn nicht, kann ein solches Gespräch vor Beginn der Programmierung Zeit sparen und bietet dem Designer und dem Entwickler die Möglichkeit, eng zusammenzuarbeiten, um Lösungen zu finden, die das gewünschte Ergebnis erzielen, ohne die Benutzerfreundlichkeit der Webseite zu beeinträchtigen.

1.1.3 Besserer Code durch das Verständnis der Grundlagen des visuellen Designs

Das Verständnis einiger Grundlagen des visuellen Designs bietet Entwicklern einige weitere Vorteile, die über eine bessere Zusammenarbeit und Kommunikation mit ihren Designkollegen hinausgehen. Der erste Vorteil, der sich direkt auf den Entwickler auswirkt, besteht darin, dass das Verständnis dafür, wie verschiedene Elemente zusammenpassen, zu einem besser organisierten und saubereren Code führen kann, der auf lange Sicht leichter zu warten ist.

Das Aufschlüsseln und Verstehen, wie eine ganze Seite aufgebaut ist, kann sich auf die Art und Weise auswirken, wie der Code geschrieben wird, um ein saubereres responsives Erlebnis mit weniger Hacks für die Programmierung für kleinere Bildschirme zu erreichen. Entwickler können auch die Stellen identifizieren, an denen eine Komponente oder ein Layout auf kleineren Bildschirmen nicht

funktioniert oder unzureichend ist (zum Beispiel, wenn eine Schaltfläche auf einem Mobiltelefon nicht groß genug ist), und sind in der Lage, visuelle Anpassungen vorzunehmen oder vorzuschlagen.

Auf einer umfassenderen Ebene wirkt sich das Verständnis der Wiederverwendung einzelner Elemente wie Überschriften und Schaltflächen in mehreren Komponenten der Benutzeroberfläche auch darauf aus, wie der Code gruppiert wird, insbesondere in einer CSS-Datei. Die Identifizierung dieser wiederverwendbaren Teile kann dazu führen, dass mehr Code zwischen den Komponenten geteilt wird, was wiederum dazu führt, dass dem Endbenutzer weniger Code zur Verfügung gestellt wird. Das ist immer ein Gewinn für die Performance des Webs. Das Design kann auch Entscheidungen über den Backend-Code beeinflussen. Einige Datenbanken müssen vielleicht von einem bestimmten Formular aus abgefragt werden oder es müssen Daten gespeichert werden. Wenn man die Benutzerströme in diesen Szenarien versteht, kann diese Arbeit früher beginnen, anstatt auf die Übergabe des Designs zu warten. Sowohl die Kommunikation als auch die Projektgeschwindigkeit verbessern sich, was wiederum allen zugutekommt.

1.1.4 Besserer Code (und Design) durch geringere Abhängigkeit von Frameworks von Drittanbietern

Viele Entwickler, die ich kenne, verwenden Komponentenbibliotheken oder Frontend-Frameworks von Drittanbietern, um viele der für ein Projekt benötigten Elemente der Benutzeroberfläche schnell zu implementieren. Sie sind schnell eingerichtet und einsatzbereit, und wenn Sie Ihren Designfähigkeiten als Entwickler nicht trauen, sind alle Teile vorhanden, um eine bessere Erfahrung zu bieten als das, was Sie glauben, selbst liefern zu können. Das Verwenden dieser Bibliotheken von Drittanbietern hat jedoch sowohl aus der Design- als auch aus der Codeperspektive mehrere Nachteile.

In den meisten Fällen erhalten Sie eine Code-Bibliothek, die mehr Code enthält, als Ihre Webseite tatsächlich nutzen wird. Außerdem handelt es sich nicht um Code, den Sie selbst geschrieben haben. Daher ist möglicherweise nicht klar, wie die verschiedenen Komponenten der Benutzeroberfläche miteinander verwoben sind oder wiederverwendet werden. Der Versuch, Code zu entfernen oder zu ändern, von dem Sie glauben, dass Sie ihn nicht verwenden werden, kann zu Fehlern in der Benutzeroberfläche führen. Außerdem sind Sie jetzt von den Autoren des Frameworks abhängig, die dafür sorgen, dass der Code des Frameworks mit neuen Webstandards und Best Practices auf dem neuesten Stand ist.

Mit dem Vertrauen und dem Verständnis für visuelle Designprinzipien und deren Anwendung auf das Web durch Layout- und UI-Komponenten können Entwickler ihr eigenes Komponentensystem aufbauen und dem Endbenutzer minimalen Code zur Verfügung stellen. Diese Codebasis lässt sich leichter pflegen, iterieren und um neue Komponenten erweitern, und die Entwickler sind nicht an ein bestimmtes Ökosystem gebunden, das sich ihrer Kontrolle entzieht.

Frameworks von Drittanbietern bieten einige sehr gängige Muster für ganze Webseiten-Layouts. In der Regel gibt es ein Layout mit einem großen Heldenbild oben und horizontalen Bändern zur Aufteilung des Seiteninhalts und ein anderes mit einem großen Bild und drei Inhaltsspalten darunter, wie in Abbildung 1–2.

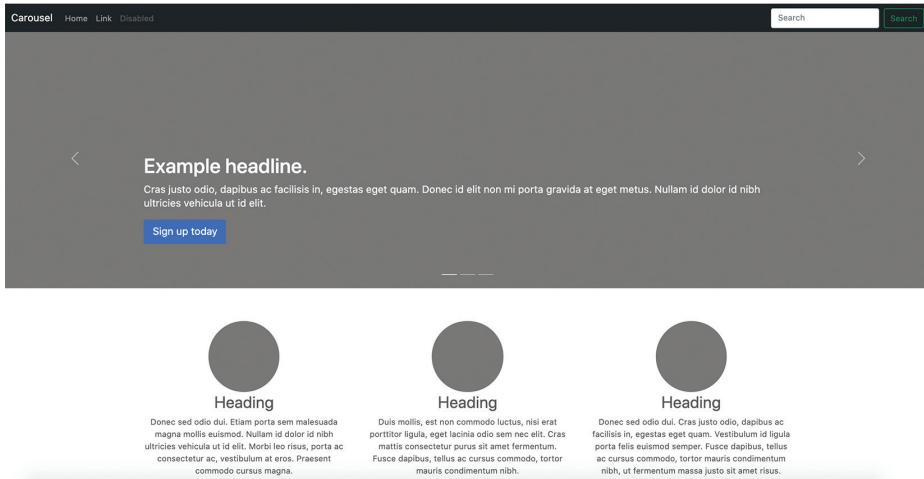


Abb. 1–2 Ein Beispiel für eine der üblicherweise von Drittanbieter-Frameworks bereitgestellten Vorlagen

Diese Layouts sind nicht per se schlecht. Sie erfüllen eine Funktion und einen Zweck. Aber sie sind oft nur auf die Haupt-Landingpage ausgerichtet, sodass Sie am Ende vielleicht eine Homepage haben, die gut gestaltet ist, aber mit Unterseiten, die nicht ganz so ausgefeilt wirken. Sie sind auch unglaublich gewöhnlich, und wenn Sie möchten, dass sich Ihre Website ein wenig mehr abhebt, werden Sie mit diesen Vorlagen nicht viel erreichen.

Die Entwicklung einer gut gestalteten Webseite oder Webanwendung ist viel mehr als nur die Auswahl einer guten Vorlage für das Layout. Das Verständnis von Designprinzipien, wie die Anwendung der Farbtheorie und die Verwendung von Typografie und Raum innerhalb eines Layouts, kann zu einem viel besseren Erlebnis für die Nutzer führen. Das bedeutet auch, dass Sie nicht versuchen müssen, den Inhalt Ihrer Website in die generische Vorlage eines anderen Anbieters zu pressen, sondern dass Sie wissen, wie Sie die Struktur Ihrer Website für Ihre Nutzer aufbauen und anpassen können. Das Layout und die Abläufe eines E-Commerce-Shops sind ganz anders als die einer Nachrichten-Website oder eines Blogs. Aber wir leben in einer sehr schnelllebigen Welt, und je nach Kunde haben Entwicklungsprojekte einen engen Zeitplan, und Sie haben vielleicht keine Wahl zwischen der Verwendung einer Vorlage und dem Erstellen Ihrer Website von Grund auf. Wenn Sie wissen, wie Sie Farbe, Typografie und Raum nutzen können, können Sie diese Kenntnisse auf alle Vorlagen anwenden, die Sie verwenden müssen, und so das Design erheblich verbessern.

1.1.5 User Experience und Entwicklung

Design als Ganzes ist ein umfangreiches Thema, mit dem man sich auseinander setzen muss. Es gibt so viele verschiedene Bereiche, auf die man sich spezialisieren kann, und User Experience ist vielleicht einer der umfangreichsten Teile des Webdesigns, der in Spezialgebiete wie Benutzerforschung und Interaktionsdesign unterteilt werden kann. Ein Großteil der User Experience konzentriert sich auf den Designaspekt des Fachgebiets, aber jede Entscheidung, die ein Entwickler trifft, ist auch eine Entscheidung für die User Experience. Einige Aspekte des User-Experience-Prozesses sind nicht unbedingt mit der eigentlichen Codeentwicklung einer Webseite verbunden. Die Kenntnis der Grundlagen kann Entwicklern jedoch helfen, die für die Benutzer getroffenen Entscheidungen zu verstehen und technische Entscheidungen zu treffen, die ihnen am meisten nützen.

Die sich stets weiter entwickelnde Rolle des Entwicklers bedeutet, die Zielgruppe zu verstehen und mit ihr in Kontakt zu treten, um eine Lösung zu entwickeln, die ihren Bedürfnissen entspricht. Methoden der User Experience können Entwicklern eine Grundlage bieten, um zu verstehen, wie sie ihre Benutzer in den Vordergrund stellen können. Auch dies kann sich auf die Art und Weise auswirken, wie Entwickler ihren Code schreiben oder welche Tools sie einsetzen, um ihr Ziel zu erreichen.

Nehmen wir an, ein Entwickler in einem kleineren Unternehmen arbeitet an einem Produkt, z.B. einer Webseite oder einer Webanwendung, und will eine neue Funktion einführen. Da es sich um ein kleines Unternehmen handelt, verfügt es weder über einen engagierten Nutzerforscher noch über das Budget, um formelle Nutzertests durchzuführen. Daher plant es, die Funktion ohne Benutzertests freizugeben und zu sehen, wie die Funktion genutzt wird. Dennoch muss ein Plan für Tests vorhanden sein, damit dieses Experiment erfolgreich ist. Das Verständnis dafür, wie man ein Design testet und Feedback sammelt, um eine Funktion zu verbessern, ist für die Produktentwicklung entscheidend.

Jede Entscheidung, die ein Entwickler trifft, wirkt sich auf die Benutzerfreundlichkeit der Webseite oder Anwendung aus. Das Verständnis der Bedeutung der verschiedenen Bereiche der User Experience und wie sie in den Lebenszyklus des Projekts passen, kann Entwicklern dabei helfen, nutzerzentrierter zu denken und Probleme mit einem Benutzererlebnis zu identifizieren, bevor zu viel Zeit in die Entwicklung investiert wird. Wenn beispielsweise Details und Funktionsmerkmale in der Prototyping-Phase ausgebügelt werden, wird sichergestellt, dass die Funktionen wie erwartet funktionieren und weniger Fehler im Produktionscode der Webseite oder Webanwendung auftreten. Das führt wiederum zu einer besseren User Experience.

User Experience und Webentwicklung sind eng miteinander verwoben. Da Entwickler immer mehr über das Unternehmen und seine Kunden wissen müssen, ist es von Vorteil, die Grundlagen der User Experience zu verstehen und zu wissen, wie sie mit der Entwicklungsphase zusammenhängen. Letztlich können diese Informationen Entwicklern dabei helfen, besseren Code zu schreiben, Code mit

weniger Fehlern auszuliefern und ihre Benutzer bei technischen Entscheidungen im Auge zu behalten. Wenn dies richtig gemacht wird, wirkt sich das Verständnis der Grundlagen der User Experience nicht nur positiv auf die endgültigen Geschäftsziele aus, sondern gibt den Entwicklern auch die Werkzeuge an die Hand, um schnell umzuschwenken und zu iterieren, wenn etwas nicht wie erwartet funktioniert, bevor es negative Auswirkungen hat.

1.2 Der Weg zu einem besseren Verständnis von Design und User Experience

Sowohl visuelles Design als auch User Experience sind extrem breit gefächerte Themen mit vielen einzelnen Fachgebieten, die eine tiefe Spezialisierung ermöglichen. Bei der Benutzererfahrung konzentrieren sich einige Funktionen auf die Designaspekte wie Wireframing und Interaktionsdesign, während andere sich ausschließlich auf die Forschung konzentrieren. Es gibt viele verschiedene Wege und Bereiche, auf die man sich spezialisieren kann, und manche verbringen Jahre damit, die Fähigkeiten für diese Spezialisierungen zu entwickeln. Jeder beginnt jedoch mit den grundlegenden Prinzipien in jedem Bereich, auf die wir in diesem Buch eingehen werden – im Kontext des Webs.

Die Kernbereiche für visuelles Design konzentrieren sich auf Komposition, Layout, Farbe sowie Typografie und wie all diese Dinge zusammenwirken, um die visuelle Hierarchie einer Seite zu schaffen. Wir werden uns mit den Grundlagen des Layouts befassen, z.B. mit den Prinzipien der Symmetrie, und uns mit Layouts befassen, die auf von Forschern ermittelten allgemeinen Lesemustern basieren. In Bezug auf Farben werden wir uns mit den Grundlagen des Farbkreises und dem Erstellen einer Farbpalette beschäftigen. Bei Tools, die eine Farbpalette für Sie erstellen, gibt es oft keine Anleitung, wie man diese Farben auf einer Website anwendet, was zu übersättigten bunten Webseiten führen kann, die nicht so raffiniert wirken wie andere.

Was die User Experience betrifft, so bestehen ganze Karrieren aus verschiedenen Spezialgebieten. Aber wir werden uns mit der Informationsarchitektur und der Definition des Benutzerflusses einer Website befassen. Was die User Experience angeht, so ist das Verständnis der verschiedenen Arten von Benutzertests und der Möglichkeiten, Feedback zu sammeln, auch ohne Zugang zu einem umfangreichen Budget bei den Entwicklern, mit denen ich zu tun habe, sehr gefragt. Wie man Feedback in ein Projekt integriert und weiß, wann etwas gut genug ist, um es zu testen, ist eine weitere häufige Frage, die sich Entwicklern stellt.

1.2.1 Der Gestaltungsprozess, den dieses Buch behandelt

Der Designprozess in diesem Buch ist auf einen iterativen Prozess ausgerichtet. Ein iterativer Prozess ist ein fortlaufender Zyklus, bei dem ein Produkt – in unserem Fall eine Webseite oder eine Webanwendung – veröffentlicht und durch schritt-

weise Änderungen verbessert wird. Dieser Prozess funktioniert jedoch auch mit Projekten, die nicht fortlaufend sind. Es gelten die gleichen Phasen, sie sind nur nicht kontinuierlich. Die einzelnen Phasen sind in Abbildung 1–3 dargestellt.

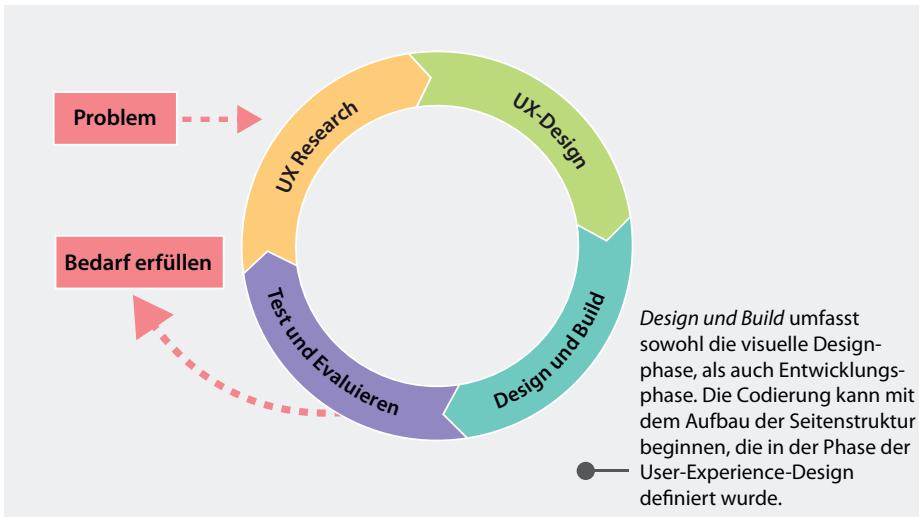


Abb. 1-3 Der iterative Prozess, den dieses Buch für ein Webdesign- oder Webanwendungsprojekt abdeckt: von der Identifizierung des Problems über die Nutzerforschung, das User-Experience-Design, die Design- und Build-Phase bis hin zu Test und Evaluierung

Die erste Phase, die ich vorstellen werde, ist die UX-Forschungsphase, in der wir von einem Kunden ein Problem genannt bekommen, das wir lösen sollen. Sie würden also eine Website oder Webanwendung erstellen, die dieses Problem löst. Bei Ihrer anfänglichen Recherche sollten Sie herausfinden, welche Anforderungen der Nutzer an Ihre Website stellt und welche Zielgruppe er anspricht. Je nachdem, was Sie entwickeln, kann diese Phase auch bedeuten, dass Sie sicherstellen, dass das Problem, das Sie lösen wollen, tatsächlich ein Nutzerproblem ist, das gelöst werden muss.

Phase 2 ist das User-Experience-Design. Sie nehmen die Kunden- und Geschäftsanforderungen für Ihr Projekt, kombinieren diese mit den Bedürfnissen, die Sie in Ihrer Benutzerforschungsphase ermittelt haben, und beginnen, einen Plan für Ihre Website zu erstellen. In dieser Phase legen Sie das Fundament für die visuellen Elemente, die Sie in der nächsten Phase einsetzen werden. Der Schwerpunkt liegt auf der Gestaltung des Seitenlayouts, der Platzierung der Inhalte und des Seitenflusses (wie die Benutzer von Punkt A zu Punkt B gelangen).

Phase 3 umfasst die Entwurfs- und die Erstellungsphase. Diese beiden Phasen bestehen aus der Anwendung visueller Designelemente wie Farben, Typografie und Plänen für Animationen, die Sie einbinden möchten, bzw. dem eigentlichen Entwicklungsprozess mit Code. Ich habe diese Phasen absichtlich zusammen-

gefasst, weil die meisten modernen Teams, die ich kenne, nicht mehr so isoliert arbeiten wie früher. Die Entwicklung kann beginnen, nachdem die Struktur der Website in der Phase des User-Experience-Designs festgelegt wurde. Visuelle Entscheidungen wie Farben, Bilder und Typografie können hinzugefügt und angepasst werden, während diese Entscheidungen getroffen werden.

In Phase 4 kommt der iterative Teil des Prozesses zum Tragen. Sie wollen, dass das, was Sie erstellt haben, die Bedürfnisse Ihrer Nutzer effizient erfüllt. Wenn dies nicht der Fall ist, müssen Sie den Prozess neu starten, um herauszufinden, warum dies nicht der Fall ist, und von dort aus Verbesserungen vornehmen. Je nach Art des Projekts, an dem Sie arbeiten, können Sie ständig neue Funktionen hinzufügen und in Iterationen aktualisieren. Bei kleineren Projekten oder freiberuflichen Projekten, z.B. einer einmaligen Marketing-Website, werden Sie die Website höchstwahrscheinlich nur einführen und sich dann davon überzeugen, dass sie funktioniert und alle Geschäftsanforderungen erfüllt. Wenn Sie die Website an einen Kunden übergeben, kann dieser die Test- und Evaluierungsphase verwalten, um sicherzustellen, dass die Website wie gewünscht funktioniert, und falls nicht, kann er den Prozess erneut starten, um Anpassungen vorzunehmen.

Die Phasen dieses Zyklus können in zwei Kategorien unterteilt werden: (a) Grundlagen der User Experience, die sich auf die Recherche, den Aufbau des Projektgerüsts (wie es aufgebaut und verbunden ist) und das Testen konzentrieren, und (b) Grundlagen des visuellen Designs. Diese visuellen Elemente werden auf das in der Benutzererfahrungsphase aufgebaute Gerüst angewendet, um dem Erlebnis durch Farben, Abstände, Typografie und Animationen Charakter und Hierarchie zu verleihen.

Grundlagen der User Experience

Die User Experience ist nicht nur ein Teil des Projekts. Oft wird sie mit der Designphase in Verbindung gebracht, doch geht sie über diese Phase hinaus. Geschäftsanforderungen, Design und Webentwicklung kommen alle zusammen, um die User Experience einer Website oder Anwendung zu schaffen. In diesem Buch geht es vor allem darum, die Bedeutung und den Zweck der verschiedenen Phasen zu verstehen: Forschung, Planung und Wireframing, die ich in den ersten Kapiteln erläutere, sowie Prototyping und Testen, und wie diese mit der Webentwicklung zusammenhängen, die in den letzten Kapiteln des Buches behandelt werden.

Die Benutzererfahrung kann sogar noch weiter in eine Untergruppe von Erfahrungen unterteilt werden, z.B. die Developer Experience, bei der es darum geht, wie ein Entwickler mit Ihrem Produkt interagiert. Die Szenarien werden jedoch viel nischiger und spezifischer. Wenn Sie die Grundlagen der User-Experience-Methodik kennen, können Sie besser verstehen, wie Sie Ihre Website oder Anwendung für die verschiedenen Nutzer entwickeln können und welche anderen Erfahrungen für Ihre Website oder Anwendung infrage kommen, wenn Ihr Zielpublikum nicht nur aus Verbrauchern besteht.

Zwar wirkt sich jede Entscheidung bei der Webentwicklung letztlich auf die User Experience aus, doch werde ich einige der spezifischsten technischen Entscheidungen anreißen, die ein Entwickler treffen muss und die die visuelle Erfahrung der Benutzer mit der Website oder Anwendung verändern können. Ich werde auch auf Entscheidungen eingehen, die sich auf die Erfahrung von Benutzern auswirken können, die nicht visuell interagieren. Die Kenntnis dieser Entscheidungen und ihrer Auswirkungen wird Entwicklern helfen, ihr Verständnis für die Zielgruppe ihres Produkts weiterzuentwickeln und ein besseres Erlebnis zu schaffen, indem sie sich in diese Zielgruppe einfühlen.

Grundlagen der visuellen Gestaltung

Design ist eine visuelle Kommunikationssprache, die man lernen kann. Es ist ein Weg, ein Problem zu lösen oder Informationen durch visuelle Mittel zu vermitteln. Wenn Sie die Grundlagen verstehen, haben Sie die Möglichkeit, Designentscheidungen zu treffen und die Gründe für diese Entscheidungen zu kommunizieren. Oft höre ich von Entwicklern, wenn es um Design geht, dass sie das Gefühl haben, nicht gut darin zu sein, weil sie entweder nicht kreativ sind oder zwar ein gutes Design erkennen können, aber nicht wissen, wie sie es selbst umsetzen können.

Bevor wir überhaupt anfangen können, unsere Typografie und Farben auszuwählen und zu überlegen, wie wir die Dinge anordnen, müssen wir mit unseren grundlegenden Designprinzipien beginnen. Dazu gehören Dinge wie Nähe, Kontrast, Symmetrie und Ausrichtung. Das Verständnis dieser Prinzipien ist vor der Phase der User Experience von entscheidender Bedeutung, weshalb das nächste Kapitel mit diesen Designprinzipien beginnt. Sie bilden die Grundlage für das Herstellen von Beziehungen zwischen den Inhalten auf der Seite, für den Aufbau von Hierarchien und für die Gestaltung von Inhalten, die Besucher auf die Seite locken. Sobald Sie diese Prinzipien kennen, können Sie getrost in die Phase des User-Experience-Designs gehen, bevor Sie zur visuellen Designphase übergehen, die Farbe, Typografie und Animation umfasst.

Wenn ich Entwickler frage, womit sie am meisten zu kämpfen haben, kommen immer wieder ein paar verschiedene Bereiche des visuellen Designs zur Sprache: Farbtheorie und die Auswahl einer Farbpalette und deren Anwendung auf eine Website oder Webanwendung, Typografie und Layout – insbesondere die Frage, was ein gutes Layout ausmacht. Im nächsten Kapitel werde ich die Designgrundlagen im Zusammenhang mit einer Website oder Webanwendung untersuchen und in späteren Kapiteln die einzelnen Designelemente. Die Fähigkeit, diese verschiedenen Designelemente (Farbe, Schrift und Raum) zusammenhängend zu verwenden und sie zu mischen, um die Hierarchie einer Seite oder Komponenten auf einer Seite zu erstellen, kann erlernt werden und hängt nicht davon ab, ob Sie sich für einen kreativen Typ halten.

1.2.2 Designexperten vs. smartes Design

Wie die Webentwicklung ist auch das Design ein Bereich mit vielen Wegen und Spezialisierungen. Dieses Buch ist nicht dazu gedacht, eine formale Ausbildung oder sogar eine informelle Ausbildung zu ersetzen, wie die Bootcamps, die viele Designer besuchen. Es ersetzt auch nicht die Erfahrung, die man durch die Arbeit im Designbereich gesammelt hat. Einige Designexperten arbeiten in Nischenbereichen wie Typografie und Animation, die durchaus zu ganzen Rollen werden können, während andere Komponentenbibliotheken und Designsysteme pflegen.

Dieses Buch wird Sie nicht zu einem Designexperten machen. Es wird Ihnen nicht das gesamte Wissen über die verschiedenen Bereiche von Design, Logos und Branding vermitteln und es wird auch nicht tief in die verschiedenen Bereiche von Design und User Experience im Zusammenhang mit dem Web eintauchen. Es ist kein Trick, um Entwickler in Designer zu verwandeln. Design und Webentwicklung sind zwei sich ergänzende Berufe, die sich häufig überschneiden. Genauso wie es für Designer, die für das Web entwerfen, von Vorteil ist, einige Kenntnisse über Code zu haben, glaube ich, dass es für Entwickler ebenso von Vorteil ist, einige Kenntnisse über Design zu haben.

1.2.3 Das Ganze zusammenfügen

In den folgenden Kapiteln lernen Sie die grundlegenden Prinzipien des Grafikdesigns im Kontext des Webs kennen und erfahren, wie Sie die verschiedenen Prinzipien nutzen können, um die Hierarchie einer Website aufzubauen. Anschließend erfahren Sie etwas über den User-Experience-Prozess, wo das visuelle Design seinen Platz hat und wie sich dieses Wissen mit dem Webentwicklungsprozess deckt. Das wird Ihre Arbeit als Entwickler verbessern.

Zusammenfassung

- Wenn Sie den Designprozess verstehen und wissen, wie Sie die Grundlagen der User Experience und des visuellen Designs anwenden, können Sie selbst besser aussehende Weberlebnisse schaffen, sich stärker in den Prozess einbringen, besser kommunizieren und sogar besseren Code schreiben.
- Wenn Sie die Grundlagen kennen und wissen, wie sie sich auf die User Experience auswirken, können Sie aus der Sicht eines Entwicklers verstehen, warum bestimmte Designentscheidungen getroffen werden.
- Die Kenntnis der Designterminologie macht Sie zu einem besseren Kommunikator mit Ihrem Team oder einem Kunden. Wenn Sie mit einem Team zusammenarbeiten, können Sie frühzeitig in das Projekt einbezogen werden, um Entscheidungen zu treffen, die die Zahl der Projektabbrüche aufgrund von Missverständnissen oder des Unverständnisses eines Designers, dass ein Entwurf nicht möglich ist, verringern.

- Anstatt sich auf Frameworks von Drittanbietern für ein Designsystem zu verlassen, können Sie Designprinzipien verwenden, um ein Designsystem zu erstellen, das die Anforderungen Ihres Projekts erfüllt und schlanker ist. Weniger Code bedeutet eine bessere Leistung im Browser, was sich positiv auf die Benutzererfahrung auswirkt.
- Die Grundlagen des visuellen Designs beginnen mit den grundlegendsten Konzepten zum Erstellen der Hierarchie und der Beziehungen zwischen den Inhalten. Es ist wichtig, diese Konzepte vor der Phase des User-Experience-Designs zu verstehen, damit Sie genau wissen, wie Sie Beziehungen zwischen den Inhalten herstellen können.
- Die Grundlagen der User Experience konzentrieren sich auf die Erforschung und Validierung der Benutzerbedürfnisse, die dann in die Phase des User-Experience-Designs einfließen. Diese Phase bildet die Grundlage, auf der die visuellen Designelemente angewendet werden.
- Visuelle Designkonzepte, wie Typografie und Farbe, sind die Teile der visuellen Designphase, die ein Design zum Leben erwecken. Sie werden auf das Fundament aufgesetzt, und Sie können die im nächsten Kapitel vorgestellten grundlegenden Designprinzipien weiter anwenden, um das Design einer Website zu verbessern.

2 Grundlagen des Designs

Dieses Kapitel behandelt

- Die Gestaltungsgrundsätze Nähe, Ausrichtung, Wiederholung, Kontrast und Ausgewogenheit
- Anwendung der Gestaltungsprinzipien, um Beziehungen zwischen den Elementen herzustellen und eine Hierarchie auf der Seite zu schaffen
- Warum es wichtig ist, diese Designgrundlagen zu verstehen, bevor man mit dem Wireframe beginnt

Das Layout ist der wichtigste Teil des Designs einer Website. Es ist buchstäblich das Fundament für ihre übrigen Teile. Betrachten wir für einen Moment den Bau eines Hauses und nicht den einer Website. Bevor Sie mit der Auswahl von Fenstern und Jalousien oder sogar der Art der Badewanne beginnen können, brauchen Sie Pläne für die Struktur und den Grundriss des Hauses. Die Größe der Badewanne kann sicherlich die Entscheidungen beeinflussen, die Sie bezüglich der Struktur des Hauses treffen. Aber Sie können keine Badewanne einbauen, bevor die anderen grundlegenden Elemente gebaut sind. Es muss eine Art Bauplan geben, der die Struktur definiert und einen Plan festlegt. Ohne einen Plan besteht die Möglichkeit, dass die Dinge einfach nach Belieben gebaut werden, und dann könnte es sein, dass die Fenster und Türen ganz unterschiedlich groß oder die Räume seltsam geformt sind. Das führt dazu, dass sich das Haus irgendwie unpassend anfühlt. Es könnte schwierig sein, sich zurechtzufinden, oder es ist unangenehm anzuschauen. Auch nach der Fertigstellung des Hauses kann es sein, dass einige Dinge noch immer nicht richtig zum Haus passen. Es fühlt sich unfertig oder ungeschliffen an. Dasselbe gilt auch für das Schreiben von Code mit HTML, CSS und JavaScript: Er muss einer bestimmten Struktur entsprechen, damit er richtig funktioniert.

Nehmen wir an, nebenan wird ein Haus gebaut, das gut durchdacht ist und einen Plan hat. Die Türen haben alle die gleiche Größe, und die Fenster sind zwar unterschiedlich groß, aber nicht alle unterschiedlich. Sie haben die gleichen Proportionen und sehen aus, als würden sie zusammenpassen. Die Räume sind nicht seltsam geformt, und die Struktur ist klar definiert. Insgesamt fühlt es sich einfacher an, sich in diesem Haus zurechtzufinden, als in dem Haus nebenan, das ohne Plan gebaut wurde. Wenn beide Häuser gleichzeitig zum Verkauf stünden, welches würde sich Ihrer Meinung nach schneller oder für mehr Geld verkaufen lassen? Wahrscheinlich das Haus, das nach einem Plan gebaut wurde und eine gut definierte Struktur hat.

Ähnlich verhält es sich mit dem Layout einer Website, vor allem wenn diese ein Produkt oder eine Dienstleistung verkaufen soll. Stellen Sie sich vor, ein Nutzer landet auf Ihrer Webseite und ist auf den ersten Blick erst einmal überwältigt. Vielleicht gibt es zu viele gleich große Inhalte, die Navigation ist unübersichtlich, oder er kann einfach nicht erkennen, was der nächste Schritt auf der Webseite sein soll. Höchstwahrscheinlich wird er Ihre Webseite wieder verlassen. Wenn Sie etwas verkaufen wollen, bedeutet dies einen Umsatzverlust, weil der Benutzer Ihre Website verlässt, bevor er die Kassenseite erreicht. In manchen Fällen wühlen sich die Benutzer durch eine schlecht gestaltete Website und legen Artikel in ihren Einkaufswagen, sind dann aber so frustriert, dass sie die Website verlassen, weil die Seite mit dem Einkaufswagen-Layout zu schwer zu verstehen ist.

Die grundlegenden Gestaltungsprinzipien, die in diesem Kapitel behandelt werden, sind entscheidend für gutes Design. Diese Leitprinzipien schaffen ein Gefühl der Ordnung durch Ausrichtung, geben dem Benutzer durch Wiederholung Orientierung, fördern die Hierarchie durch Kontrast und geben dem Design durch Ausgewogenheit ein Gefühl der Mitte. Nähe, Ausrichtung, Wiederholung, Kontrast und Ausgewogenheit im Kontext des Layouts einer Website helfen Ihnen, eine solide Grundlage für das Design Ihrer Website zu schaffen. Sobald Sie die Hauptstruktur festgelegt haben, werden diese fünf Grundsätze beim Hinzufügen der kleineren Komponenten wichtig. Sie sind alle für jeden Teil einer Website relevant, vom Hauptlayout bis hin zur Gruppierung von Text und Schaltflächen, und können miteinander kombiniert werden, um ein Design interessanter zu gestalten.

Es ist wichtig, diese Grundsätze zu verstehen, bevor Sie die Phase des visuellen Designs erreichen, noch bevor Sie mit dem Erstellen eines Wireframes in der User-Experience-Phase beginnen. Wireframes sind die Blaupausen oder Schemata für Ihre Website, die ihre Skelettstruktur definieren, in der Sie beginnen, die verschiedenen Inhalte der Website zu gruppieren und anzugeordnen. Hier entsteht das ursprüngliche Layout einer Website, und die erste Anwendung dieser Prinzipien kommt ins Spiel, wenn wir Inhalte platzieren und Beziehungen zwischen den einzelnen Teilen herstellen. In den nächsten Kapiteln werden Sie sich eingehender mit dem Wireframing beschäftigen.

2.1 Die Grundsätze der Gestaltung

Obwohl Design ein weites Feld ist, das sich in viele verschiedene Disziplinen und Bereiche aufteilen lässt, auf die man sich spezialisieren kann – vom Druck- und Verpackungsdesign bis hin zum Design von Videospielen –, gibt es einige wenige Grundlagen, die für alle Bereiche gelten. Wenn Sie diese Prinzipien verstehen, legen Sie den Grundstein für visuelles Design und verstehen, warum Designer einige ihrer Entscheidungen treffen und wie Sie diese Prinzipien anwenden können, wenn Sie keinen Zugang zu einem Designer haben. Man muss kein Designexperte sein, um eine Website gut aussehen zu lassen oder Designänderungen vorzunehmen; diese Grundsätze sind die Grundlage dafür, wie man diese Dinge tut.

Je nachdem, mit wem Sie sprechen, ist die Anzahl der Gestaltungsgrundlagen unterschiedlich. Für die Zwecke dieses Buches und um die absoluten Grundlagen herauszuarbeiten, werde ich fünf Grundlagen erörtern: Nähe, Wiederholung, Ausrichtung, Kontrast sowie Gleichgewicht und Symmetrie. Keines dieser Elemente ist unbedingt wichtiger als ein anderes, und sie werden selten allein verwendet. Sie sind alle Teil eines zusammenhängenden Systems, um die visuelle Hierarchie einer Seite zu schaffen und das Auge durch den Inhalt zu führen. Diese Grundlagen gelten nicht nur für das Internet, sondern werden auch im Druck- und Verpackungsdesign verwendet. Da wir aber Webdesigner sind, werden wir jedes dieser Elemente im Kontext des Webs bewerten.

2.1.1 Nähe

Nähe ist das Gestaltungsprinzip von Beziehungen. Je näher die Elemente beieinander liegen, desto mehr werden sie als verwandt oder als Teil derselben Gruppe angesehen. Die Anwendung von Nähe, um verschiedene Elemente zusammenzufassen, ist im Grunde die Art und Weise, wie wir Informationen organisieren. Wenn Elemente näher beieinander liegen, nehmen wir sie als Teil einer Gruppe oder als miteinander verbunden wahr. Nehmen Sie die beiden unterschiedlichen Reihen von Kreisen in Abbildung 2–1.

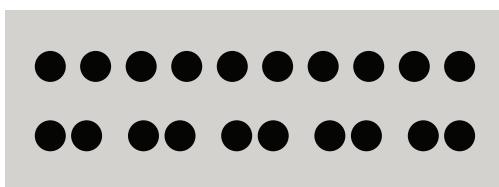


Abb. 2-1

Durch die Abstandsanpassung zwischen den Formen entstehen in der zweiten Zeile kleinere Zweiergruppen anstelle einer großen Gruppe von Kreisen.

Die erste Reihe der Kreise ist gleichmäßig verteilt, sodass alle Kreise als Teil derselben Gruppe wahrgenommen werden. Wenn wir in der zweiten Reihe zwei Kreise näher aneinanderrücken und den Abstand zwischen den einzelnen Paaren vergrößern, entstehen durch die Nähe der beiden Kreise in jedem Paar fünf sepa-

rate Gruppen. Die Verwendung von Leerraum zwischen den Elementen ist der Schlüssel zur Gruppierung von Inhalten im Web, um eine sinnvolle Unterscheidung für Ihre Benutzer zu schaffen. Mit leicht identifizierbaren Inhaltsgruppen können Benutzer Ihre Website oder Anwendung leichter durchsuchen. Elemente, die sich auf denselben Abschnitt oder dieselbe Komponente beziehen oder Teil desselben Abschnitts sind, sollten zusammen gruppiert werden und auf dem Bildschirm nahe beieinander erscheinen. Das Prinzip der räumlichen Nähe ist der Schlüssel zur Einteilung von Inhalten in Gruppen. In Abbildung 2–2 sind die Elemente zwar unterschiedlich geformt, aber es ist dennoch klar, dass es zwei Gruppierungen gibt.

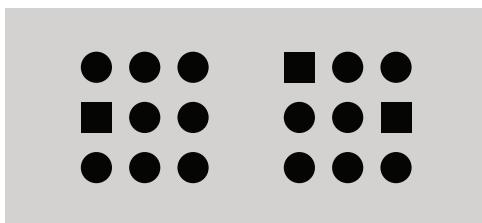


Abb. 2-2

Obwohl aus verschiedenen Formen bestehend, sind es aufgrund des großen Abstands zwei unterschiedliche Gruppen.

Wenn wir die Anwendung anderer Prinzipien ignorieren und uns nur auf die Nähe zur Unterscheidung verlassen würden, könnten wir immer noch erkennen, welche Elemente zusammengehören und eine Einheit bilden. Betrachten Sie Abbildung 2–3 als eine ungestylte Liste von Blogpost-Titeln mit einem einleitenden Textausschnitt unter jedem Titel.

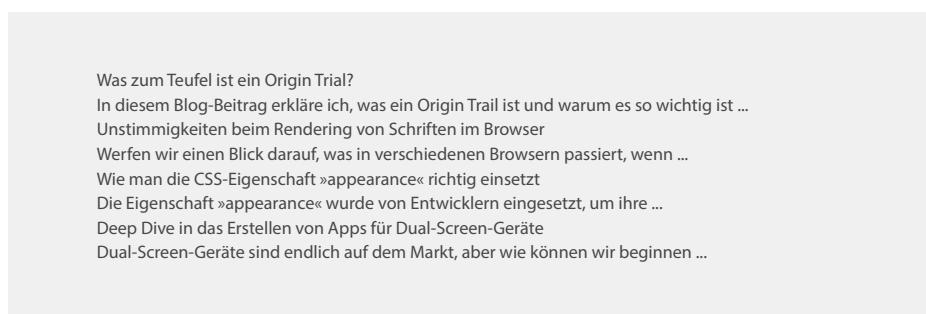


Abb. 2-3 *Eine Liste von Blogpost-Titeln und eine Vorschau des Blogpost-Texts ohne Styling, Abstände oder eine klare Hierarchie*

Ähnlich wie die gleichmäßig verteilte horizontale Reihe von Kreisen ist auch diese vertikale Liste von Blogpost-Titeln mit den dazugehörigen Artikelvorschauen gleichmäßig verstreut. Auf den ersten Blick scheinen diese Elemente Teil derselben Gruppe zu sein. Erst bei näherer Betrachtung erkennt ein Betrachter das Gegen teil und ist wahrscheinlich sogar ein wenig verwirrt von dem Inhalt, der ihm

präsentiert wird. Aufgrund der gleichmäßigen Abstände zwischen den einzelnen Elementen scheint die Liste eine einzige große Gruppe zu sein. Nun vergrößern wir den Leerraum unter jeder Textvorschau wie in Abbildung 2–4, ohne den Raum unter dem Titel des Blogbeitrags zu verändern, um die einzelnen Elemente in Paaren zu gruppieren. Sobald wir etwas Platz zwischen den einzelnen Titelpaaren und Textausschnitten hinzufügen, werden unsere Gruppen deutlicher. Wenn wir nun die Liste durchsuchen, wird deutlich, dass es aufgrund des Leerraums zwischen den einzelnen Gruppierungen vier separate Inhaltsabschnitte gibt.

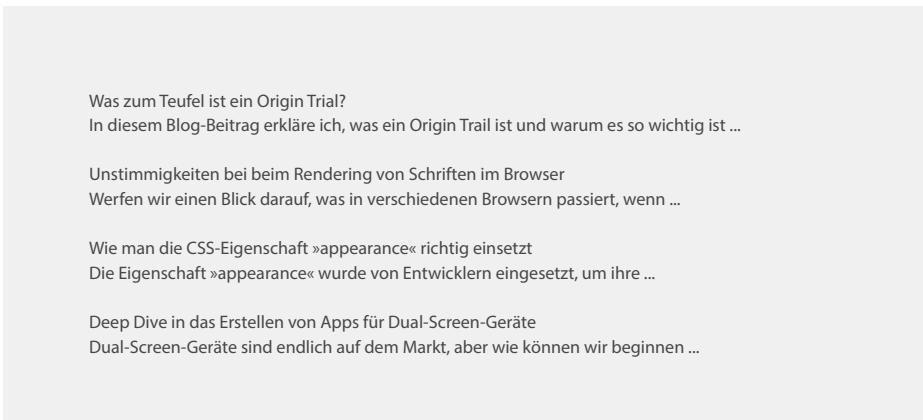


Abb. 2–4 *Blogtitel und Vorschautexte sind in Paaren gruppiert und durch Leerraum getrennt, um die Beziehung zwischen den einzelnen Inhalten zu verdeutlichen. Jede Gruppierung ist ein separater Teil.*

Leerraum im Design ist nichts, wovor man sich fürchten muss. Er ist äußerst wichtig, um ein Design zu schaffen, das für die Benutzer Ihrer Website leicht zu konsumieren ist. Viele Designanfänger wollen so viel Platz wie möglich auf einer Seite füllen. Außerdem habe ich oft von Kunden die Rückmeldung erhalten, dass es zu viel Leerraum gibt, und sie wollen alles auffüllen, indem sie Elemente größer machen und den Abstand zwischen ihnen verringern. Im Web wird die Benutzerfreundlichkeit beeinträchtigt, wenn zusammengehörige Elemente nicht angemessen gruppiert sind, was das Überprüfen von Inhalten erschwert. Schauen wir uns zum Beispiel die Liste der Blogbeiträge noch einmal an, ändern aber den Stil des Titels, um ihn deutlicher zu machen (Abb. 2–5).

Die Hervorhebung des Blogpost-Titels trägt zwar zu einer gewissen Unterscheidung bei, aber es gibt immer noch keine klare Beziehung zwischen den Elementen, da alles gleichmäßig verteilt ist. Die zusätzliche Hervorhebung des Blogpost-Titels reicht nicht aus, um zu verdeutlichen, dass es sich um vier getrennte Inhaltsgruppen handeln soll. Fügen wir also wieder Leerraum unter jeder nicht hervorgehobenen Textzeile in Abbildung 2–6 hinzu, um unsere Inhaltsgruppen zu erstellen.



Abb. 2-5 Der Titel des Blogeintrags wird nun in fetter Schrift hervorgehoben, sodass er sich von dem darunter stehenden Text abhebt.

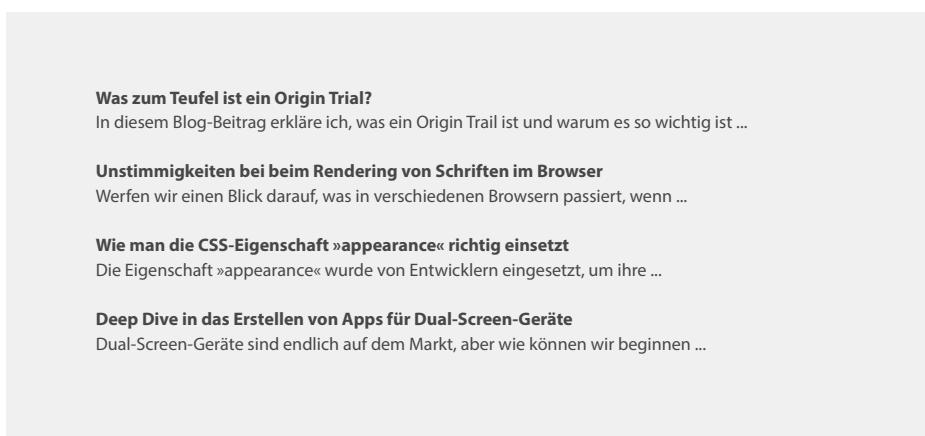


Abb. 2-6 Durch das Hinzufügen von Leerraum lassen sich Inhaltsgruppen bilden und klarere Beziehungen zwischen den Inhalten herstellen. Die fett gedruckte Überschrift zeigt, dass sie von dem darunter liegenden Text getrennt ist, aber die Nähe zur darunter liegenden Zeile zeigt, dass sie immer noch Teil derselben Gruppe ist.

Verwechslungen vermeiden, indem nicht zusammengehörige Teile getrennt gehalten werden

In den meisten Fällen versuchen die Nutzer, die auf Ihre Website kommen, eine Aufgabe zu erledigen, und sind daher darauf bedacht, das Gewünschte so schnell wie möglich zu finden. Es ist wichtig, daran zu denken, dass einige Elemente auf Ihrer Seite zwar ähnlich aussehen und auf die gleiche Weise funktionieren, dies aber nicht immer bedeutet, dass sie zusammen gruppiert werden sollten.

Daher sollte sorgfältig überlegt werden, was zu einer Gruppierung gehört. Wenn Sie z.B. einen Stil für Schaltflächen auf Ihrer Website haben und eine Ihrer

Seiten zwei unterschiedliche und nicht miteinander verbundene Gruppen von Schaltflächen enthält, z.B. Navigationsschaltflächen und Schaltflächen, die eine Aktion ausführen (z.B. Löschen oder Bearbeiten), stellen Sie sicher, dass zwischen den beiden Gruppen genügend Platz ist. Mischen Sie die Schaltflächen nicht miteinander, da dies zu einer verwirrenden Erfahrung für Ihre Benutzer führen kann, die keine zusätzliche Zeit damit verbringen wollen, über die verschiedenen Aktionen nachzudenken, die die Schaltflächen ausführen können (Abb. 2–7 und 2–8).

The screenshot shows a user interface titled "Add files". Below the title, there are three separate groups, each containing a checkbox and the text "File name.pdf". Each group is enclosed in a light gray rectangular box. At the bottom of the interface, there is a horizontal row of five buttons: "Next", "Back", "Add", "Edit", and "Delete".

Abb. 2–7 In diesem Teil der Benutzeroberfläche sind alle Schaltflächen für die Navigation im Formular zusammen gruppiert und sehen ähnlich aus. Der Benutzer braucht Zeit, um die gesuchte Aktionsschaltfläche zu finden.

The screenshot shows a user interface titled "Add files". Below the title, there are three separate groups, each containing a checkbox and the text "File name.pdf". Each group is enclosed in a light gray rectangular box. At the bottom of the interface, there is a horizontal row of five buttons: "Next", "Back", "Add", "Edit", and "Delete".

Abb. 2–8 Eine bessere Lösung besteht darin, die Aktionsschaltflächen und die Navigationsschaltflächen auf verschiedenen Seiten der Seite anzurufen.

Leicht durchsuchbare Websites nach dem Prinzip der Nähe erstellen

Die Nähe ist auch wichtig, wenn es darum geht, wie Sie Ihre Inhalte gruppieren und anordnen, um eine leicht zu lesende Website zu gestalten. Sehen Sie sich ein Beispiel für eine Seite an, auf der die Elemente nicht gruppiert sind, sondern nur gleichmäßig voneinander getrennt sind (Abb. 2–9).

In diesem Beispiel ist alles auf der Seite gleichmäßig verteilt, und die Nähe wird nicht dazu verwendet, Inhalte zusammenzufassen. Wir wissen also nicht, ob sich das obere Rechteck auf den darunter liegenden Text bezieht oder ob sich das untere Rechteck auf den darüber oder darunter liegenden Text bezieht.

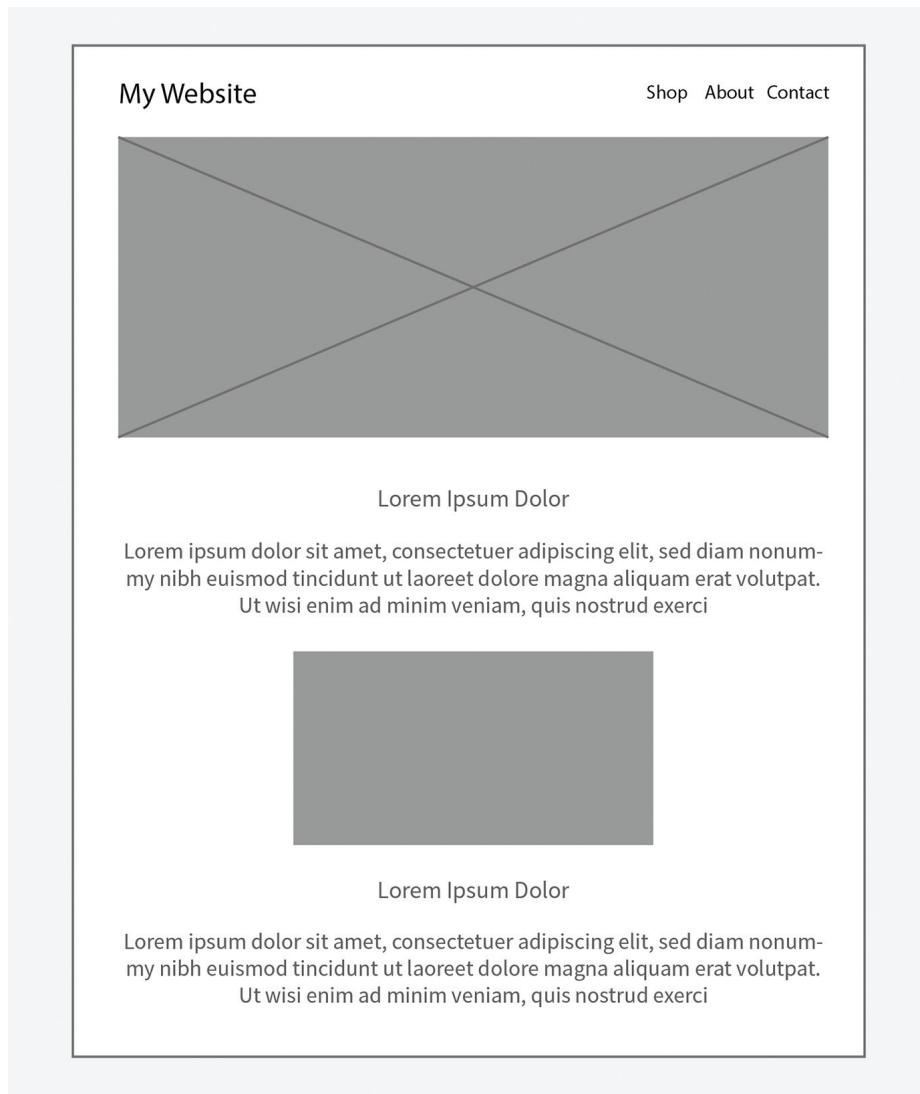


Abb. 2-9 In diesem Beispiel sind die Abstände zwischen den einzelnen Inhalten gleichmäßig, sodass wir nicht wissen, ob die Bilder (die Rechtecke) mit dem Inhalt darunter oder darüber in Beziehung stehen.

In Abbildung 2–10 nutze ich das Prinzip der Nähe, um zusammengehörige Elemente näher aneinander zu rücken und Gruppen zu bilden, die auf der Seite unterscheidbar sind. Das obere Rechteck steht nicht in direktem Zusammenhang mit dem darunter liegenden Inhalt, aber das untere Rechteck wurde näher an den ersten Textblock gruppiert, und der Text wurde näher an die zugehörige Überschrift verschoben, damit wir jede einzelne Inhaltsgruppe identifizieren können.

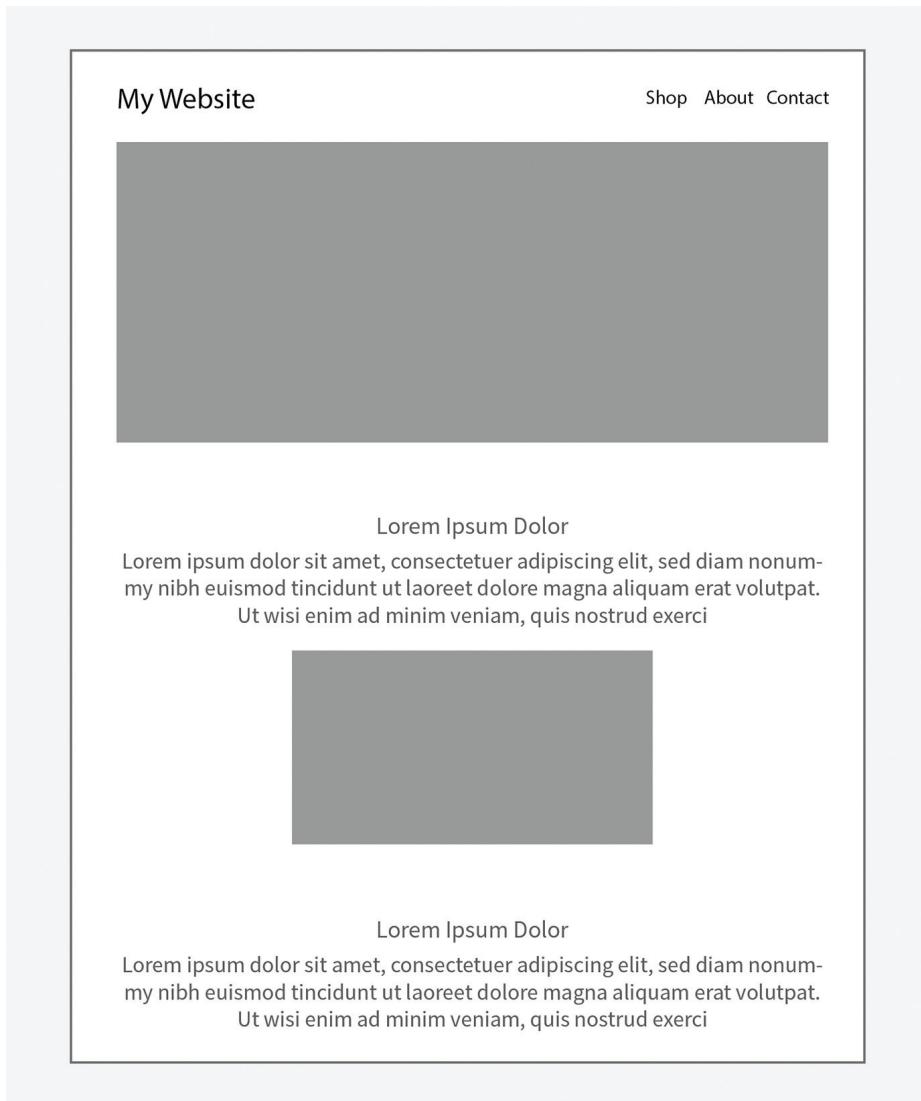


Abb. 2-10 In diesem Beispiel stelle ich Beziehungen zwischen den Inhalten her, indem ich verwandte Inhalte auf der Seite gruppieren und Abschnitte erstelle.

Sehen Sie sich ein Beispiel von der Website *webhint.io* in Abbildung 2-11 an. Wenn wir den gesamten Rand und das Padding aus dem Inhalt entfernen, erhalten wir einen Inhaltsblock auf der rechten Seite, der schwer zu erfassen ist. Alle Teile des Inhalts liegen übereinander und sind eng beieinander. Einige Überschriften sind zwar größer, aber die Beziehung zwischen den verschiedenen Zeilen des Inhalts ist nicht klar.

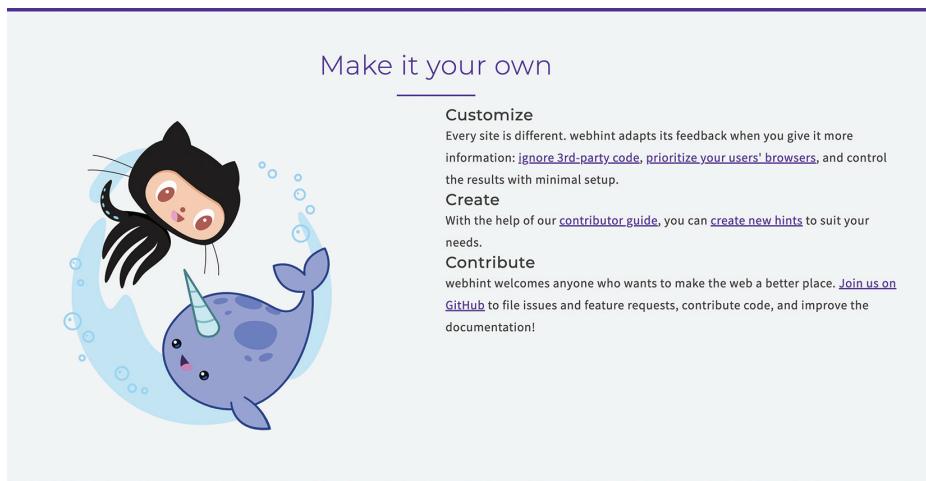


Abb. 2-11 In diesem Beispiel wird weder Leerraum noch räumliche Nähe genutzt, sodass eine Wand aus Text entsteht.

Durch die Verwendung von Nähe, wie in Abbildung 2–12, schaffen wir jedoch Gruppen von Inhalten, indem wir die Textblöcke näher an ihre jeweiligen Überschriften rücken und den Abstand zwischen den einzelnen Gruppen vergrößern. Die Seite wird viel übersichtlicher und besser lesbar.

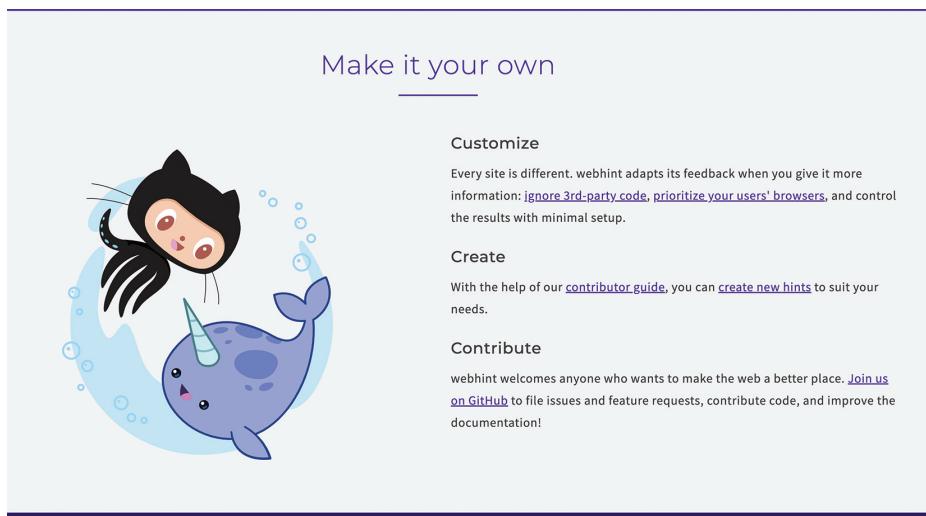


Abb. 2-12 Nachdem die Überschriften und Textblöcke gruppiert sind, lassen sich die einzelnen Inhaltsgruppen leichter durchsuchen und erfassen.

Wenn Sie das Prinzip der Nähe auf die verschiedenen Elemente der Benutzeroberfläche auf Ihrer Seite anwenden, besteht das Ziel darin, für Ihre Benutzer leicht erkennbare Inhaltsbereiche zu schaffen. Sie wollen die kognitive Belastung reduzie-

ren und eine Benutzeroberfläche schaffen, die nicht überwältigend ist. Formulare sind ein weiterer Bereich, in dem Nähe dazu beitragen kann, dass die Benutzer nicht das Gefühl haben, auf eine Wand von Inhalten zu starren. Abbildung 2–13 zeigt ein Bestellformular mit zeilenweise angeordneten Eingabefeldern.

The screenshot displays a horizontal form layout with the following fields:

- First Name:** Rose
- Last Name:** Tyler
- Company (Optional):** (empty field)
- Email Address:** badwolf@email.com
- Country / Region:** United States
- Address 1:** 11 Blue Box Lane
- Address 2:** (empty field)
- Zip/Postal:** 21102
- City:** Manhattan
- State:** New York
- Phone Number:** 908-555-5309
- Card Number:** 4573XXXXXXXXXXXX
- Security Code:** (empty field)
- Name on Card:** Rose Tyler
- Expiration Date:** 05 - May 2016

Abb. 2–13 Dieses Kassenformular besteht nur aus einer Reihe von Eingabefeldern, und die benötigten Informationen sind nicht klar.

Optisch ist es ein wenig überwältigend. Wenn wir uns den Code ansehen, finden wir eine Zeile nach der anderen mit HTML-Texteingaben und gelegentlich einem HTML-<select>-Element. Aber wenn wir das Formular in verschiedene Abschnitte unterteilen, wie in Abbildung 2–14, wird es weniger überwältigend. Wir erstellen zusammengehörige Gruppen innerhalb des Formulars, die dem Benutzer einen natürlichen Anhaltspunkt bieten, wenn er eine Pause einlegen und später zurückkehren muss, um das Formular auszufüllen.

The image displays two screenshots of a form interface. The top screenshot shows the 'Billing Information' section, which contains fields for First Name (Rose), Last Name (Tyler), Company (Optional) (empty), Email Address (badwolf@email.com), Country / Region (United States), Address 1 (11 Blue Box Lane), Address 2 (empty), Zip/Postal (21102), City (Manhattan), State (New York), and Phone Number (908-555-5309). The bottom screenshot shows the 'Payment Information' section, which contains fields for Card Number (4573XXXXXXXXXXXX), Security Code (empty), Name on Card (Rose Tyler), and Expiration Date (05 - May, 2016).

Abb. 2–14 Wenn Sie zusammengehörige Eingabefelder gruppieren und Abschnittsüberschriften hinzufügen, wird das Kassenformular beim Scannen einfacher zu analysieren.

Diese Gruppierung sorgt auch für eine klarere Organisation des Frontend-Codes. Die beiden Abschnitte Rechnungs- und die Zahlungsinformationen sind im Code in eigenen `div`-Containern gruppiert. Statt einer riesigen Liste von HTML-Eingaben wird der Code auch optisch so formatiert, dass er zwei getrennte Abschnitte anzeigt. Das verringert die kognitive Belastung des Entwicklers, wenn er den Code durchsucht, um Änderungen vorzunehmen.

Ziel ist es, dass der Benutzer weniger darüber nachdenken muss, wie er mit Ihrer Seite interagieren kann. Der erste Schritt auf dem Weg zu einer übersichtlichen Benutzeroberfläche besteht darin, durch die Gruppierung von Elementen klar identifizierbare Beziehungen herzustellen.

Nähe: Das Prinzip der Beziehungen

Nähe ist das Prinzip der Beziehungen und vielleicht die wichtigste Grundlage, die es zu verstehen gilt. Es kommt ins Spiel, wenn Sie Ihre Website planen und organisieren, bevor Sie überhaupt anfangen, über visuelles Design und Stil nachzudenken. Wenn Sie die Wireframing-Phase eines Projekts erreichen, sollten Sie die Präzision und die Festlegung von visuellen Designdetails ignorieren. Konzentrieren Sie sich stattdessen darauf, Inhalte und Komponenten zu gruppieren und zu organisieren und die grobe Platzierung der Inhalte auf der Website zu bestimmen, damit die Beziehungen zwischen den größeren Inhaltsblöcken sinnvoll sind. Achten Sie auf Leerraum zwischen nicht zusammenhängenden Inhaltsgruppen, um die Seite für die Benutzer übersichtlicher zu machen und Verwirrung oder Unsicherheit zu vermeiden, wenn sie sich auf das Erledigen einer Aufgabe konzentrieren.

Die Gruppierung zusammengehöriger Inhalte, wenn Sie keine anderen Gestaltungsprinzipien auf Ihre Inhalte angewendet haben, ist die einfachste und grundlegendste Möglichkeit, damit Sie zwischen Inhaltsgruppen unterscheiden können. Das Beispiel in Abbildung 2–4 zeigt dieses Konzept auf der einfachsten Ebene, mit nichts als Leerraum zwischen den einzelnen Blogtiteln und Artikelschnipseln. Das Definieren von Beziehungen zwischen Inhalten durch Nähe, bevor andere Prinzipien angewandt werden, hilft beim Aufbau der Seitenstruktur und stellt sicher, dass die Beziehungen klar sind, bevor die entsprechenden Prinzipien angewandt werden, die eine weitere Hierarchie im Layout einer Website schaffen.

2.1.2 Ausrichtung

Bei dem Begriff Ausrichtung denken viele Entwickler an verschiedene CSS-Eigenschaften und an die Positionierung von Elementen auf einer Seite mit CSS. Vielleicht erinnern sich auch viele an eines der häufigsten Layout-Probleme in der Geschichte des Webs: die vertikale und zentrierte Ausrichtung von Text auf einer Webseite oder in einem `<div>`. Die Ausrichtung im Web bezieht sich auf die Positionierung von Elementen entlang verschiedener Achsen, die Ausrichtung von Elementgruppen auf eine bestimmte Weise, um eine zusammenhängende Einheit zu bilden, und die Textausrichtung, z.B. rechtsbündig oder im Blocksatz.

Bei der Gestaltung für das Web ist die Ausrichtung der Elemente etwas starrer als beim Design von Druck-Erzeugnissen. In der Software, die wir für das Design von Druck-Erzeugnissen verwenden, gibt es keine Einschränkungen. Im Web, in einem Browser, sind wir durch die Webplattform und den Code, den wir zur Definition unseres Layouts schreiben, eingeschränkt. Es ist schwieriger, ein Element willkürlich auf einer Webseite zu platzieren, als dies bei der Gestaltung von Drucksachen der Fall ist. Selbst bei Drag-and-Drop-Layout-Editoren für Webseiten ist das Ziehen und Ablegen auf den Gesamtumriss der vordefinierten Webseite beschränkt. Sie können beliebige Elemente einfügen, aber Sie können sie nicht zufällig platzieren.

Als Entwickler sind Sie damit vertraut, wie man Code verwendet, um verschiedene Elemente auf der Seite auszurichten. Ein gewisses Maß an Struktur für die Ausrichtung ist bereits über die Webplattform eingebaut, wenn Sie etwas wie CSS Grids oder Flexbox verwenden, um das Layout Ihrer Seite zu definieren. Wenn ich zum Beispiel `set display: flex;` für ein übergeordnetes Element mit zwei untergeordneten Elementen einstelle, hat die Webplattform bereits einen Plan dafür, wie dieses Layout mit nur dieser Codezeile dargestellt wird. Wenn Sie jedoch eine Webseite entwerfen, werden Sie `display: flex;` höchstwahrscheinlich nicht allein in einer Zeile verwenden. Hier ist es wichtig, das Prinzip der Ausrichtung zu verstehen, denn die Ausrichtung sorgt für Ordnung auf der Seite.

Die Ausrichtung ist eine weitere Möglichkeit, Elemente auf einer Seite zu organisieren und in Beziehung zueinander zu setzen, indem sie an einer unsichtbaren Linie ausgerichtet werden. Diese Ausrichtung trägt zur Strukturierung einer Seite bei und ist auch ein Schlüssel zur Lesbarkeit. Selbst wenn Sie ein Layout für eine Webseite erstellen, das von den heute üblichen Mustern abweicht und etwas willkürlicher und unstrukturierter erscheint, kann die Ausrichtung bestimmter Elementgruppen dazu beitragen, dass ein zufälliges Layout ausgefeilt und strukturiert erscheint.

Verschiedene Arten von Ausrichtung

Es gibt zwei Hauptmöglichkeiten, um Elemente auf einer Seite auszurichten: die Ausrichtung am Rand und die Ausrichtung in der Mitte. Bei der Ausrichtung am Rand werden die Elemente entlang des Randes einer Komposition ausgerichtet und können am oberen, unteren, linken oder rechten Rand ausgerichtet werden, wie in Abbildung 2–15 zu sehen. Eine mittige Ausrichtung liegt vor, wenn die Elemente horizontal oder vertikal an einer zentralen Achse ausgerichtet sind. Eine horizontale Ausrichtung liegt vor, wenn die Elemente durch die Mitte ihrer horizontalen Achsen ausgerichtet sind. Und eine vertikale Ausrichtung, wenn die Elemente durch die Mitte ihrer vertikalen Achsen ausgerichtet sind, wie in Abbildung 2–16 dargestellt.

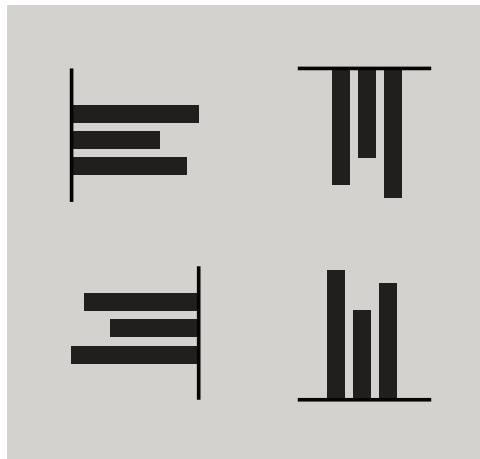


Abb. 2-15 Die verschiedenen Arten der Ausrichtung am Rand von oben links nach rechts:
linke Ausrichtung, obere Ausrichtung, rechte Ausrichtung, untere Ausrichtung

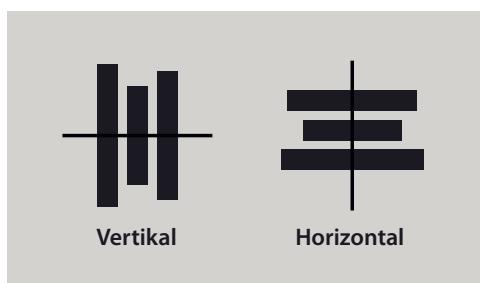


Abb. 2-16 Vertikale und horizontale Zentrierung werden im Verhältnis zu ihrer Mittelachse dargestellt.

Die Ausrichtung an den Rändern und die Ausrichtung an der Mitte gelten nicht nur für die Elementgruppen auf einer Seite. Ausrichtungswerte können auch für Text festgelegt werden, um die unsichtbare Linie zu definieren, wenn alle Elemente an derselben Sache ausgerichtet sind. Für die Textausrichtung gibt es im Web, genau wie im Druck, verschiedene Möglichkeiten. Die verschiedenen Arten der Textausrichtung sind links, rechts, zentriert und Blocksatz, wie in Abbildung 2-17 dargestellt. Während die Textausrichtung links, rechts und mittig die bereits erwähnten Konzepte der Ausrichtung am Rand widerspiegelt, ist der Blocksatz eine Art der Ausrichtung, bei der der Abstand zwischen den Wörtern entweder vergrößert oder verkleinert wird, um sowohl die linke als auch die rechte Seite des Textblocks auszurichten. Wie das Beispiel in Abbildung 2-17 zeigt, besteht das Hauptproblem bei Text im Blocksatz darin, dass sich große Lücken zwischen den Wörtern bilden können, die die Lesbarkeit beeinträchtigen, wie ich in Kapitel 7 erläutern werde.

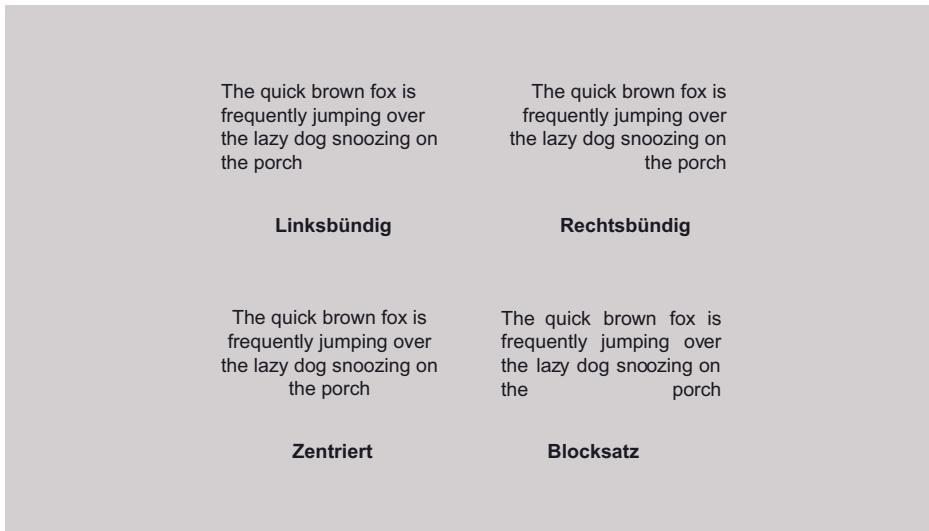


Abb. 2-17 Die verschiedenen Arten der Textausrichtung, die Sie im Layout anwenden können

Gemischte Anordnung

Auch wenn die Textausrichtung ein separater Wert ist, der bei der Ausrichtung verschiedener Elemente berücksichtigt werden muss, ist sie doch ein wichtiger Bestandteil des Seitenlayouts und trägt zur Schaffung von Ordnung und einer klaren Hierarchie bei.

In einigen Fällen können mehrere Elemente, einschließlich eines Textblocks, alle an einer Kante ausgerichtet werden. Die Ausrichtung kann jedoch uneinheitlich aussehen, da der Text und nicht sein Begrenzungsrahmen zentriert ist.

Das Mischen der Ausrichtung von Elementen auf einer Seite kann ein einzigartigeres und interessanteres Layout schaffen, aber es sollte sich mehr auf die größeren Inhaltsgruppen konzentrieren als auf die einzelnen Teile, aus denen eine Gruppe besteht. Schauen Sie sich Abbildung 2-18 als Beispiel an.

Wir haben eine Reihe von Links zu Blogbeiträgen. Jede Bloglink-Komponente besteht aus einer Überschrift und einem Ausschnitt des Artikels, gefolgt von einer Schaltfläche *Weiterlesen*. In dem Beispiel in Abbildung 2-18 sind der Blogtitel, der Artikelschnipsel und die Schaltfläche linksbündig ausgerichtet, wie an ihren Begrenzungsrahmen zu erkennen ist. Der Text des Artikelschnipsels ist jedoch mittig ausgerichtet. Betrachten Sie nun die Komponenten ohne Begrenzungsrahmen in Abbildung 2-19.

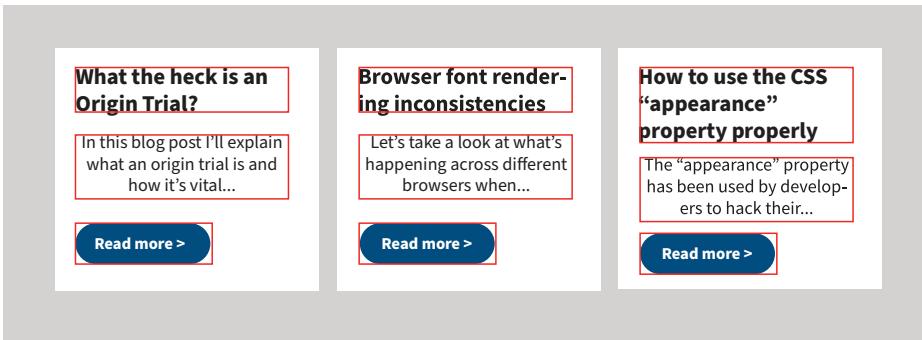


Abb. 2–18 Die Begrenzungsrahmen des Inhalts sind umrisseen, um zu zeigen, dass Elemente zwar auf eine bestimmte Weise ausgerichtet werden können, aber alle unsere `<div>`-Elemente (z.B. eine unterschiedliche Textausrichtung innerhalb der `<div>`-Elemente) einen starken Einfluss darauf haben können, wie kohärent eine Komponente wirkt.

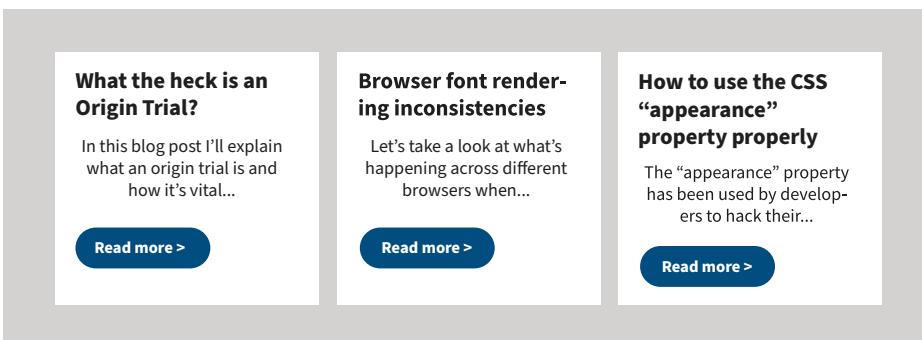


Abb. 2–19 Der Textausschnitt in der Mitte ist im Gegensatz zur Überschrift und zur Schaltfläche zentriert.

Die unsichtbaren Begrenzungsrahmen sind nicht mehr eingezeichnet, sodass Sie sehen können, dass der Text mittig ausgerichtet ist, obwohl der Begrenzungsrahmen des Blogsnipsels links ausgerichtet bleibt. Dieses Design sieht etwas unpassend aus, weil die scharfe Kante, die entsteht, wenn alle Elemente links ausgerichtet sind, durch den mittleren Text unterbrochen wird, der mittig ausgerichtet ist. Es gibt keine saubere unsichtbare Linie entlang der linken Seite, die den Inhalt verbindet. Wenn Sie die Ausrichtung des Textes in Abbildung 2–20 so ändern, dass er links ausgerichtet ist, wird die Veränderung hin zu einer kohärenteren Komponente deutlich.

Diese Diskussion soll Ihnen nicht davon abraten, die Ausrichtung von Elementen zu mischen, wenn Sie das Layout einer Webseite durchdenken. Das Mischen der Ausrichtung kann ein interessantes Layout erzeugen. Konzentrieren Sie sich also auf die Ausrichtungsentscheidungen für die Komponenten als Ganzes innerhalb von Abschnitten auf einer Webseite.

What the heck is an Origin Trial?
In this blog post I'll explain what an origin trial is and how it's vital...

Browser font rendering inconsistencies
Let's take a look at what's happening across different browsers when...

How to use the CSS "appearance" property properly
The "appearance" property has been used by developers to hack their...

Abb. 2-20 Alle Komponenten sind linksbündig ausgerichtet und wirken kohärenter.

Als ich die Website für das Linting-Tool *webhint* entwarf, durchlief die Seite viele verschiedene Aktualisierungsrunden. Ich erinnere mich, dass ich mit verschiedenen Ausrichtungsoptionen für einige der einzelnen Inhaltskomponenten gespielt habe. Ich habe versucht, links ausgerichteten Text mit Inhalten zu mischen, die insgesamt zentriert waren, aber das wirkte ungeschliffen. Stattdessen konzentrierte ich mich auf die einzelnen Abschnitte oder Inhaltsbereiche und wechselte die Ausrichtungsmethoden, um die verschiedenen Inhaltsabschnitte zu verstärken (Abb. 2-21).

Use webhint to improve your website

webhint is a customizable linting tool that helps you improve your site's accessibility, speed, cross-browser compatibility, and more by checking your code for best practices and common errors.

Hints to help you throughout the web development cycle

Develop

Use the [VS Code extension](#) to get real-time feedback in your IDE when writing code.

Debug

The [browser extension](#) lets you scan a site from within your browser's DevTools.

Deliver

Use the [webhint CLI](#) in your CI/CD pipeline to analyze your site before it goes into production.

Abb. 2-21 Dieses Beispiel zeigt eine wechselnde Ausrichtung zwischen den Abschnitten. Das trägt dazu bei, den Inhalt auf der Seite zu differenzieren und die Aufmerksamkeit zu wecken.

Wie Sie in Abbildung 2–21 sehen können, ist der erste Inhaltsbereich links ausgerichtet, und der zweite Bereich besteht aus einigen weiteren Inhaltsgruppen, die von einem Bild begleitet werden. Da die Illustrationen hier keine harte rechteckige Form wie ein Bild haben, habe ich mich dafür entschieden, den Inhalt jeder Gruppe in der Mitte auszurichten. Das Hauptziel der mittigen Ausrichtung des Inhalts in diesem Abschnitt war jedoch, ihn von dem darüber liegenden Abschnitt zu unterscheiden.

Schauen Sie sich in Abbildung 2–22 an, wie dieser Abschnitt aussieht, wenn der gesamte Inhalt links ausgerichtet ist.

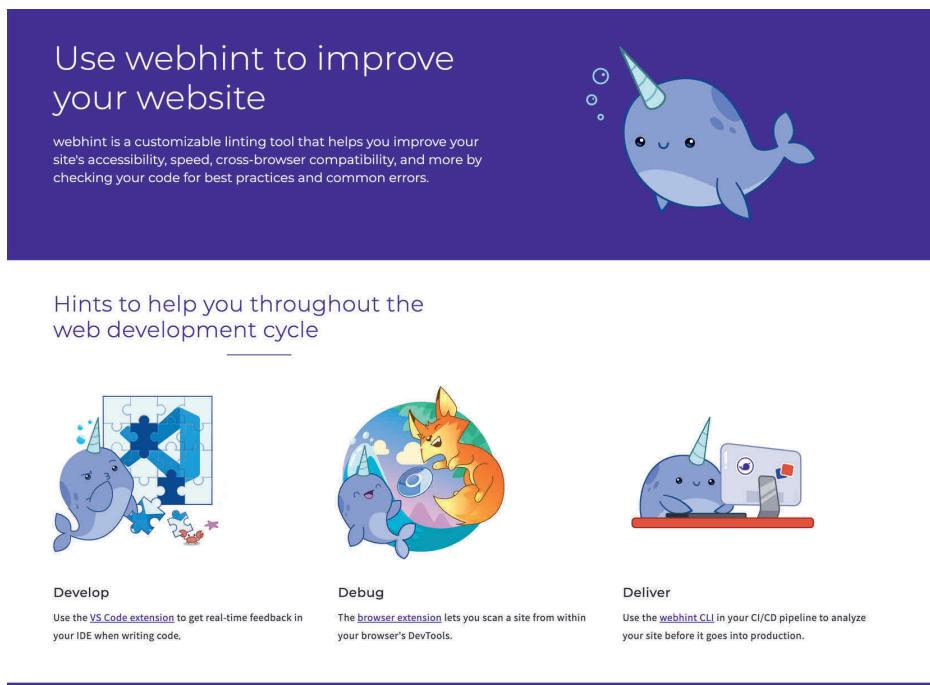


Abb. 2–22 Der zweite Abschnitt der Webseite ist wie der obere und untere Abschnitt links ausgerichtet.

Obwohl der weiße Streifen diesen Bereich als neuen Abschnitt kennzeichnet, fügt er sich in den Rest der Seite ein, wenn wir alles nach links ausrichten. Die zusätzliche Ausrichtung in der Mitte trägt dazu bei, den Unterschied zwischen den Abschnitten zu verdeutlichen und die kognitive Belastung beim Durchsuchen des Inhalts der Webseite zu verringern.

Eine gute Ausrichtung hilft dabei, unsichtbare Linien zu schaffen, die das Auge über die Seite leiten und ein Gefühl von Ordnung statt Unordnung vermitteln. In Abbildung 2–23 habe ich auf einer Hotel-Website einige Hotels aufgelistet, und obwohl sie oben ausgerichtet sind, wird der Inhalt in der rechten Spalte durch die unterschiedliche Länge des Inhalts weiter nach unten gedrückt als die beiden anderen. Auch die Namen der Hotels sind im Gegensatz zu der anderen Komponente mittig ausgerichtet. Dadurch wird die unsichtbare Linie unterbro-

chen, die auf der linken Seite durch die verschiedenen Komponenten gebildet wird, die mit Ausnahme des Hotelnamens perfekt aneinander gereiht sind.

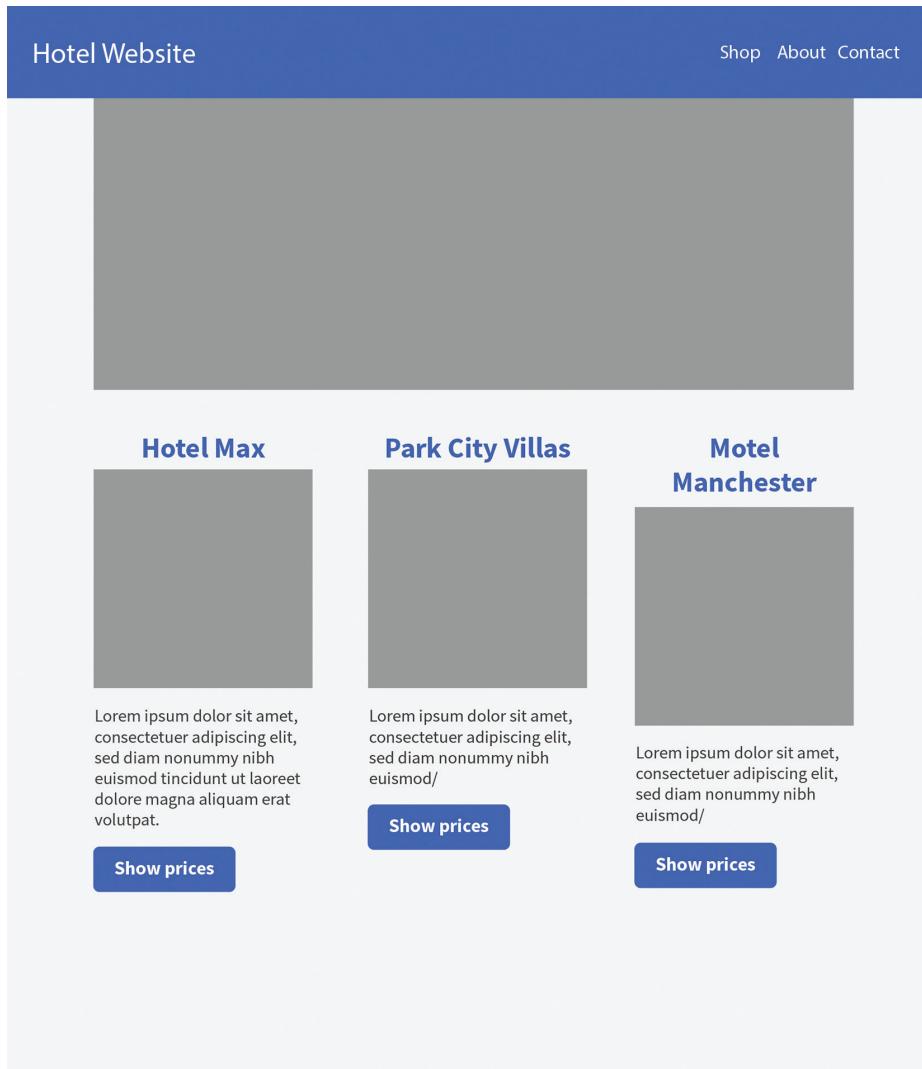


Abb. 2-23 In diesem Beispiel sind die Hotelnamen mittig ausgerichtet, und obwohl sie alle am gleichen Punkt oben ausgerichtet sind, wirkt die Seite unzusammenhängend und nicht richtig ausgerichtet.

In Abbildung 2–24 passe ich das Layout an, indem ich die Hotelnamen links ausrichte und nach unten unter die Bilder verschiebe. Die Bilder haben alle die gleiche Größe, was ein Gefühl der Ordnung schafft. Der Hotelname auf der rechten Seite erstreckt sich zwar immer noch über zwei Zeilen, aber da ich auch die Schaltflächen am unteren Rand ausgerichtet habe, haben alle Komponenten nun

die gleiche Größe. Der Inhalt in der Mitte muss nicht unbedingt perfekt ausgerichtet sein, da die oberen und unteren Komponenten dies tun, wodurch in jeder Spalte eine gleich große Form entsteht.

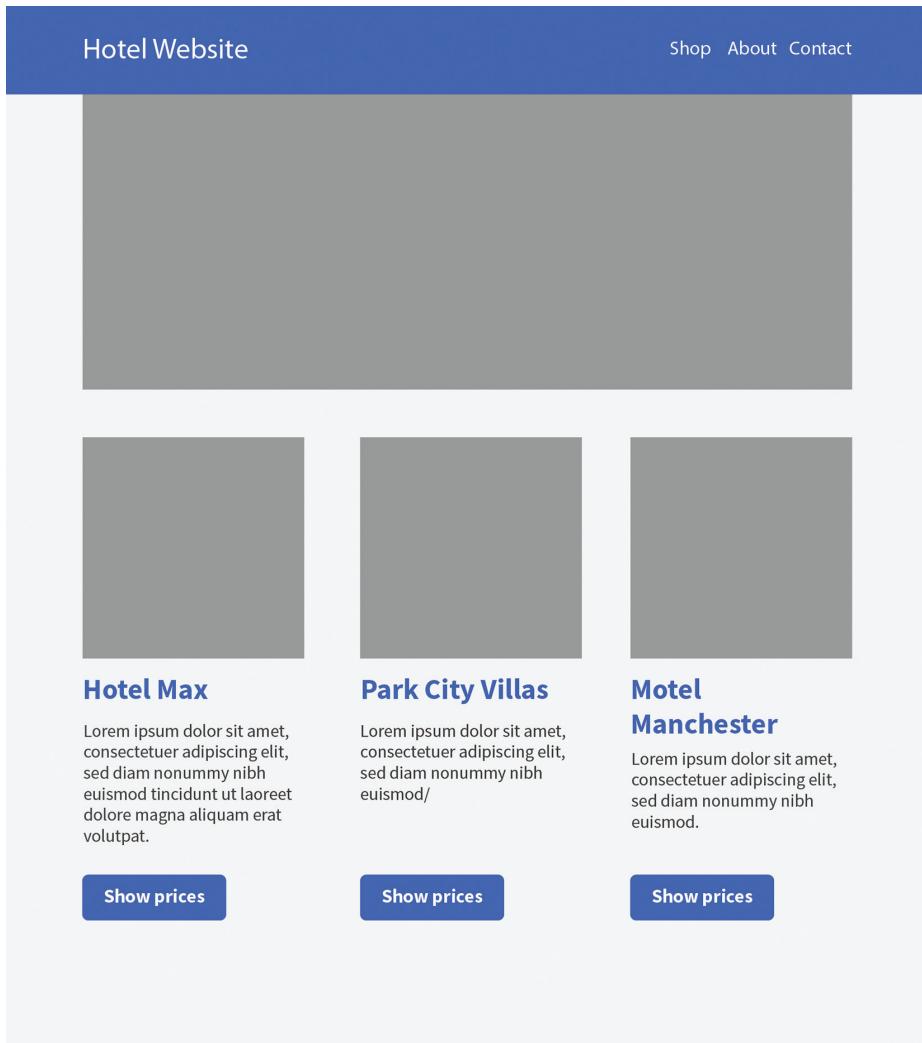


Abb. 2-24 Durch die Ausrichtung des gesamten Inhalts nach links und das Verschieben des Hotelnamens unter das Bild wird der Eindruck erweckt, dass jede Komponente ein Rechteck ist, und es entsteht ein geordneteres Bild auf der Seite.

Das zweite Beispiel ist aufgrund der unsichtbaren Linien viel übersichtlicher, die dadurch entstehen, dass der gesamte Inhalt links, oben und unten ausgerichtet ist. Manchmal passt der Inhalt nicht perfekt in eine Zeile, aber wenn die obere und untere Komponente ausgerichtet sind, muss der mittlere Inhalt nicht die gleiche Größe haben.

Inhaltsübersicht

Teil 1

Grundlagen der Gestaltung

1

1	Die Kluft zwischen Design und Entwicklung überbrücken	3
2	Grundlagen des Designs	19

Teil 2

User Experience

61

3	Grundlagen der User Experience	63
4	Nutzerforschung	77
5	User-Experience-Design	101

Teil 3

Visuelle Gestaltungselemente

123

6	Web-Layout und Gestaltung	125
7	Web-Layouts mit Animationen aufwerten	159
8	Auswahl von und Arbeit mit Typografie im Web	173
9	Farbtheorie	199
10	Aufbau einer Website	225

Teil 4	
Nach der visuellen Gestaltung	251
11 Testen, Validieren, Iterieren	253
12 Entwicklungsentscheidungen und User Experience	261

Inhaltsverzeichnis

Einleitung	xvii
Vorwort	xix
Danksagung	xxi
Über dieses Buch	xxiii
Teil 1	
Grundlagen der Gestaltung	1
<hr/>	
1 Die Kluft zwischen Design und Entwicklung überbrücken	3
1.1 Wie Entwickler von Design- und User-Experience-Grundlagen profitieren	4
1.1.1 Zusammenarbeit und Kommunikation verbessern	5
1.1.2 Die Gründe für Designentscheidungen verstehen	6
1.1.3 Besserer Code durch das Verständnis der Grundlagen des visuellen Designs	8
1.1.4 Besserer Code (und Design) durch geringere Abhängigkeit von Frameworks von Drittanbietern	9
1.1.5 User Experience und Entwicklung	11
1.2 Der Weg zu einem besseren Verständnis von Design und User Experience	12
1.2.1 Der Gestaltungsprozess, den dieses Buch behandelt	12
1.2.2 Designexperten vs. smartes Design	16
1.2.3 Das Ganze zusammenfügen	16
Zusammenfassung	16

2	Grundlagen des Designs	19
2.1	Die Grundsätze der Gestaltung	21
2.1.1	Nähe	21
2.1.2	Ausrichtung	31
2.1.3	Wiederholung	40
2.1.4	Kontrast	46
2.1.5	Ausgewogenheit	53
2.2	Designgrundlagen für die User Experience und darüber hinaus	59
	Zusammenfassung	59
Teil 2		
User Experience		61
3	Grundlagen der User Experience	63
3.1	Benutzererfahrung geht über die visuelle Gestaltung hinaus	64
3.2	Was ist User Experience?	65
3.3	Die User-Experience-Rollen	67
3.3.1	Nutzerforschung	67
3.3.2	Gestaltung der User Experience	68
3.3.3	User-Experience-Texter	69
3.4	Methoden der User Experience	70
3.4.1	Nutzerzentrierter Entwurf	71
3.4.2	Double-Diamond-Prozess	73
3.5	User-Experience-Praktiken vertiefen	74
	Zusammenfassung	75
4	Nutzerforschung	77
4.1	Einführung in die Nutzerforschung	79
4.1.1	Nutzerforschung scheint kostspielig zu sein, aber Misserfolg ist teurer	79
4.1.2	Erste Datenerhebung	80
4.1.3	User-Personas	81
4.1.4	Nutzerbedürfnisse	82
4.1.5	Ziele der Website festlegen: Abgleich von Nutzer- und Unternehmenszielen	83
4.1.6	S.M.A.R.T.-Ziele erstellen	85

4.2	Strategien der Nutzerforschung	87
4.2.1	Qualitativ vs. quantitativ	89
4.2.2	Benutzereinstellung vs. Benutzerverhalten	90
4.2.3	Methoden der Nutzerforschung	91
4.2.4	Forschung in einer kontinuierlichen Phase	98
	Zusammenfassung	98
5	User-Experience-Design	101
5.1	Informationsarchitektur	102
5.1.1	Sitemap	103
5.1.2	Bestandsaufnahme und Prüfung der Inhalte	107
5.2	User Flows und User Journeys	109
5.2.1	User Flow	109
5.2.2	User Journey	111
5.3	Gestaltung Ihrer Website und Anwendung	114
5.3.1	Wireframing	114
5.3.2	Gestaltung der Benutzeroberfläche und farbige Mock-ups	118
5.3.3	Prototyping	119
	Zusammenfassung	121

Teil 3**Visuelle Gestaltungselemente****123**

6	Web-Layout und Gestaltung	125
6.1	Aufbau der Struktur Ihrer Website	127
6.2	Ein Raster verwenden	129
6.2.1	Das Raster festlegen	129
6.2.2	Die Abmessungen des Rasters	130
6.3	Auswahl eines Layouts	132
6.3.1	Einspaltige Muster verwenden	133
6.3.2	Gängige mehrspaltige Muster	135
6.3.3	Lesemuster	142
6.4	Raum nutzen	146

6.5	Überlegungen zum responsiven Design	150
6.5.1	Für Mobilgeräte gestalten	150
6.5.2	Design für Multiscreen- und faltbare Geräte	155
	Zusammenfassung	157
7	Web-Layouts mit Animationen aufwerten	159
7.1	Warum eine Animation?	160
7.1.1	Verbesserung der Benutzerfreundlichkeit	160
7.1.2	Unvergessliche Interaktionen schaffen	160
7.1.3	Animationen als Dekoration verwenden, um eine emotionale Reaktion hervorzurufen	161
7.2	Wann sollte eine Animation verwendet werden?	163
7.2.1	Hinweise für die Navigation auf der Webseite geben	163
7.2.2	Rückmeldung an die Benutzer bei der Interaktion mit UI-Elementen geben	163
7.2.3	Navigation und Seitenübergänge	164
7.2.4	Den Status von etwas in Arbeit anzeigen	165
7.2.5	Langfristig denken	168
7.3	Wie man Animationen plant	168
7.3.1	Wie man ein Storyboard erstellt	169
7.4	Technische Aspekte der Animation	170
7.4.1	Überlegungen zur Performance	170
7.4.2	Überlegungen zur Barrierefreiheit	171
	Zusammenfassung	172
8	Auswahl von und Arbeit mit Typografie im Web	173
8.1	Die Grundlagen von Schrift	174
8.1.1	Zwischen einer Schrift und einem Schriftschnitt unterscheiden	174
8.1.2	Unterschiedliche Schriftklassifizierungen	175
8.1.3	Schriftarten und -schnitte	181
8.2	Auswahl einer Schrift für das Web	184
8.2.1	Was macht eine gute Webschrift aus?	184
8.2.2	Wie man Schriftarten und Schriften miteinander verbindet	185
8.2.3	Eine Schriftrampe aufbauen	189

8.2.4	Ihren Schriftarten Größen zuweisen	190
8.2.5	Einen vertikalen Rhythmus festlegen	191
8.2.6	Für Lesbarkeit sorgen	195
	Zusammenfassung	198
9	Farbtheorie	199
9.1	Terminologie der Farben	201
9.1.1	Farbtöne	201
9.1.2	Warme vs. kühle Farben	202
9.1.3	Farbton, Sättigung und Helligkeit	203
9.2	Der Farbkreis	205
9.2.1	Farbbeziehungen	205
9.2.2	Beispiele für Farben studieren und dann weitere Beispiele ansehen	210
9.3	Farbpsychologie	210
9.4	Auswählen und Anwenden eines Farbschemas	212
9.5	Tutorial: Wählen Sie ein Farbschema und wenden Sie es an	213
9.6	Überlegungen zur Barrierefreiheit	219
9.6.1	Farbkontrast prüfen	220
9.6.2	Verlassen Sie sich nicht nur auf die Farbe, um einen Zustand oder Status anzudeuten	220
9.7	Farbmodi im Web	221
9.7.1	RGB und RGBA	221
9.7.2	Hexadezimal	222
9.7.3	HSL und HSLA	222
9.7.4	CIE Lab und LCH	222
9.7.5	Welchen Farbmodus soll ich verwenden?	223
9.8	Farbabweichungen auf Bildschirmen	223
	Zusammenfassung	224
10	Aufbau einer Website	225
10.1	Die Anforderungen an die Projekt-Website	225
10.2	Die Platzierung von Inhalten mit einem Wireframe festlegen	226
10.3	Rastersystem und Abstände festlegen	227
10.4	Typografie auswählen	229

10.5	Vertikalen Rhythmus festlegen	230
10.6	Bildmaterial auswählen	237
10.6.1	Bilder verwenden, um den Ton anzugeben	237
10.6.2	Text auf Bildern	238
10.7	Farbpalette auswählen und anwenden	241
10.8	Letzte Handgriffe	243
10.9	Responsives Design	244
10.9.1	Tablet-Design, das achtspaltige Raster	244
10.9.2	Mobiles Design, das vierspaltige Raster	246
10.9.3	Mobile-first Design	249
	Zusammenfassung	249

Teil 4**Nach der visuellen Gestaltung****251**

11	Testen, Validieren, Iterieren	253
11.1	Der Zyklus des Designs	253
11.1.1	Ein bestehendes Design ersetzen	254
11.1.2	Minimum Viable Product	254
11.1.3	Wann ist ein Entwurf gut genug, um mit dem Testen zu beginnen?	255
11.2	Testarten	255
11.2.1	Kundeninterviews	255
11.2.2	Testen in einer Produktionsumgebung	256
11.2.3	A/B-Tests	257
11.2.4	Gestaffelte Rollouts	258
11.2.5	Anfängliche Methoden der Nutzerforschung	259
11.3	Prüfergebnisse anwenden	259
	Zusammenfassung	260

12	Entwicklungsentscheidungen und User Experience	261
12.1	Die Wirkung des geschriebenen Codes	262
12.1.1	Warum es wichtig ist, gutes HTML zu schreiben	262
12.1.2	Performance und Ladezeiten	265
12.1.3	Webtechnologien ohne Cross-Browser-Unterstützung	265
12.2	Entwickler bauen für Entwickler	266
	Zusammenfassung	267
	Weitere Ressourcen	269
	Index	273