

Auf einen Blick

1	Denken in flexiblen Strukturen	21
2	Umsetzung eines fixen Designs in ein flexibles Layout	41
3	Die Schlüsseltechnologie Media Queries	57
4	Ein responsiver Workflow	89
5	Design und Typografie	115
6	Semantik und Barrierefreiheit	151
7	Responsive Layout-Patterns	169
8	Frameworks für responsives Design	217
9	Navigationskonzepte	247
10	Flexible Bildelemente	283
11	Mehr flexible Inhalte	355
12	Qualitätssicherung und Optimierung	431
13	Fazit	471

Inhalt

Geleitwort des Fachgutachters zur ersten Auflage	15
Vorwort	17

1 Denken in flexiblen Strukturen 21

1.1 Responsive Webdesign: Was bedeutet das eigentlich?	21
1.1.1 Veränderte Anforderungen an die Darstellung von Websites	22
1.1.2 Anpassungsfähige Websites versus Mobilversionen von Websites ...	25
1.1.3 Beispiele für anpassungsfähige Websites	26
1.2 Layouttypen: feste, fluide und flexible	29
1.2.1 Das feste Layout	29
1.2.2 Das fluide und das elastische Layout	30
1.2.3 Das adaptive Layout	30
1.2.4 Das responsive Layout	31
1.3 Vom fixen zum flexiblen Raster (»Grid«)	32
1.4 Adaptives Layout: festes Layout mit Umbrüchen	36
1.4.1 Gegenüberstellung: adaptives versus fixes Layout	38
1.5 Responsives Layout: fluides Layout mit Umbrüchen	38
1.5.1 Gegenüberstellung: responsives versus adaptives Layout	39
1.6 Zusammenfassung	39

2 Umsetzung eines fixen Designs in ein flexibles Layout 41

2.1 Die Ausgangslage: ein grafischer Entwurf mit festen Abmessungen	41
2.2 Der erste Schritt: feste Raster in flexible umrechnen	47
2.3 Der zweite Schritt zu mehr Flexibilität: anpassungsfähige Inhalte	49
2.4 Der dritte Schritt: Layouts mit Media Queries umschalten	52
2.5 Zusammenfassung	55

3	Die Schlüsseltechnologie Media Queries	57
3.1	Cascading Stylesheets (ein kurzer Rückblick)	58
3.1.1	Zuweisung von CSS-Eigenschaften	58
3.2	Medientyp (media type)	59
3.2.1	Medienabfrage für den Medientyp setzen	59
3.2.2	Medientypen in der Übersicht	60
3.3	Medieneigenschaften (media features)	61
3.3.1	Medieneigenschaften in der Übersicht	62
3.4	Media Queries schreiben	64
3.4.1	Komponenten eines Media Querys	64
3.5	Viewports und Pixel	66
3.5.1	Der visuelle Viewport	66
3.5.2	Der Layout-Viewport auf mobilen Geräten	66
3.5.3	Gerätepixel und CSS-Pixel: der »virtuelle« visuelle Viewport	67
3.5.4	Das Viewport-Metatag und seine Eigenschaften	70
3.5.5	Die @viewport-Anweisung in CSS	73
3.6	Media Queries im Responsive Webdesign	75
3.6.1	Medieneigenschaft width	75
3.6.2	Medieneigenschaft device-width	76
3.6.3	Media Queries in em	76
3.6.4	Medieneigenschaft height – vertikale Media Queries	78
3.6.5	Medieneigenschaft orientation	79
3.6.6	Medieneigenschaft aspect-ratio, device-aspect-ratio	80
3.6.7	Medieneigenschaft resolution und device-pixel-ratio	80
3.7	Media Queries im HTML-Header oder im Stylesheet	84
3.8	Das wichtigste Media Query ist kein Media Query!	84
3.9	Media Queries und die alten Internet Explorer	84
3.9.1	Nur Basislayout zuweisen	85
3.9.2	Mittlere Layoutstufen mit Conditional Comments zuweisen	85
3.9.3	Media-Query-Unterstützung mit JavaScript nachrüsten	85
3.10	User-Agent-Sniffing und serverseitige Erkennung	86
3.10.1	WURFL & Co	87
3.11	Zusammenfassung	87

4	Ein responsiver Workflow	89
4.1	Der alte Prozess	89
4.2	Phase 1: Moodboards und Inhaltsplan	92
4.2.1	Moodboards	92
4.2.2	Inhalte strukturieren und hierarchisieren	93
4.2.3	Content-Prototypen	95
4.3	Phase 2: Style Tiles und Wireframes	96
4.3.1	Style Tiles	96
4.3.2	Mockups	100
4.4	Phase 3: Design im Browser	104
4.4.1	Online-Editoren	105
4.4.2	Komponenten, Patterns und Atomic Design	106
4.5	Phase 4: Rinse and Repeat	107
4.5.1	Beziehen Sie Ihre Kunden in den Gesamtprozess mit ein	107
4.6	Das responsive Team	109
4.7	Dokumentation responsiver Designs	110
4.7.1	Style-Dokumentation mit Evernote	111
4.7.2	Dexy	113
4.8	Zusammenfassung	113
5	Design und Typografie	115
5.1	Design follows Content	115
5.1.1	You cannot not communicate – kein Design ist auch ein Design	116
5.2	Design von innen nach außen – der Atomic-Design-Ansatz	118
5.2.1	Atomic Design anwenden mit Pattern Lab	119
5.3	Designanforderungen für responsive Sites	122
5.3.1	Does size matter – was machen Nutzer mit ihren Geräten?	123
5.3.2	Geräteübergreifendes Surfen	125
5.3.3	Size matters: Ziele für Touch-Events	125
5.3.4	Es gibt kein Hover auf Hawaii	127
5.3.5	Inaktives :active auf iOS	128
5.3.6	Handpositionen	128
5.3.7	Layoutwechsel bei Änderung der Orientierung	129

5.3.8	Schreiben ist mühsam: Formulare auf Smartphones	132
5.3.9	Mobile Inspiration und Best Practice	132
5.4	Typografie (anpassungsfähiger Text)	133
5.4.1	Die Auswahl der Schriftart	134
5.4.2	Angaben für die Schriftgröße	137
5.4.3	Schriftgrößenwahrnehmung auf kleinen und großen Bildschirmen	141
5.4.4	Zeilenlänge und Durchschuss	144
5.4.5	Skalierbare seitenbreite Texte	148
5.5	Zusammenfassung	150
6	Semantik und Barrierefreiheit	151
6.1	Prinzipien der Zugänglichkeit	151
6.1.1	Wahrnehmbarkeit	152
6.1.2	Bedienbarkeit	157
6.1.3	Verständlichkeit	158
6.1.4	Robustheit	159
6.2	Semantik durch HTML5	159
6.2.1	section	160
6.2.2	article	161
6.2.3	nav	161
6.2.4	aside	161
6.2.5	header	161
6.2.6	footer	162
6.3	HTML5-Formularattribute für mehr Semantik	163
6.3.1	Neue Eingabeelemente	163
6.4	Semantik durch WAI-ARIA-Rollen	165
6.5	Zusammenfassung	168
7	Responsive Layout-Patterns	169
7.1	Mobile First	170
7.1.1	Reduktion auf das Wesentliche ist die Devise	170
7.1.2	Mobile First – Progressive Enhancement fürs Layout	172

7.1.3	Mobile First bedeutet auch Content First	173
7.1.4	Fallbacklayout: Was ist die »Notfallansicht«?	174
7.2	Praxisbeispiel: Mobile First	175
7.2.1	Mobile First – los geht's!	175
7.2.2	Mockups für das Layout	176
7.2.3	Basisversion: Smartphone-Ansicht	177
7.2.4	Setzen des Viewport-Metatags	184
7.3	Auswahl der Breakpoints	186
7.3.1	Haupt-Breakpoints	187
7.3.2	Anpassungs-Breakpoints	188
7.3.3	Vertikale Breakpoints	189
7.4	Praxisbeispiel: ersten Breakpoint setzen (Tablets)	190
7.5	Layout-Patterns (Darstellungsmuster) für unterschiedliche Ausgabegeräte	195
7.5.1	Tiny Tweaks (kleine Optimierungen)	196
7.5.2	Mostly Fluid (größtenteils fließend)	197
7.5.3	Column Drop (abgesenkte Spalten)	198
7.5.4	Layout Shifter (Layoutverdrehen)	199
7.5.5	Off-Canvas-Layout (außerhalb des Bildschirms)	199
7.5.6	Footer-Navigation und Off-Canvas-Marginalie	200
7.5.7	Top-Off-Canvas-Menü und Off-Canvas-Marginalie	201
7.5.8	Vertikale und horizontale Off-Canvas-Panels	201
7.5.9	Zusammengefasste Elemente Off-Canvas	202
7.5.10	Off-Canvas-Lösungen aus der Schublade	203
7.6	Praxisbeispiel: weitere Breakpoints setzen (große Screens)	204
7.6.1	Kleine Desktopversion nach dem Konzept Layout Shifter	204
7.6.2	Große Desktopversion	206
7.7	Breakpoint-Tools	208
7.7.1	Breakpoints testen mit Browsertools	208
7.7.2	Gridpak zum Erstellen von Rastern mit Media Queries	210
7.8	CSS3-Layouttechniken für responsive Layouts	211
7.8.1	Flexbox-Layout	211
7.8.2	Grid-Layout (CSS3)	215
7.9	Zusammenfassung	216

8	Frameworks für responsives Design	217
8.1	Eigene Vorlage oder fertige Frameworks verwenden?	218
8.2	Wie wählen Sie das richtige Framework aus?	220
8.3	Eine kurze Vorstellung responsiver Frameworks	221
8.3.1	YAML	222
8.3.2	Foundation	226
8.3.3	Bootstrap	230
8.3.4	PureCSS	233
8.4	JavaScript-Bibliotheken und andere Helfer	235
8.4.1	jQuery	235
8.4.2	Modernizr	236
8.4.3	Elegante Stylesheets mit Präprozessoren: SASS	238
8.5	Zusammenfassung	245
9	Navigationskonzepte	247
9.1	Was macht eine Navigation benutzerfreundlich?	247
9.2	Benutzerfreundliche Navigation für mobile Geräte	248
9.2.1	Freier Blick auf die Website	249
9.2.2	Ausreichend große Klickflächen für Touchbedienung	250
9.2.3	Umgang mit Menüs mit mehreren Ebenen	250
9.3	Navigationstypen für mobile Geräte mit Touchscreen	250
9.3.1	Wenige Menüpunkte am oberen Rand	251
9.3.2	Lange Menüs am oberen Rand	254
9.3.3	Select-Menü	255
9.3.4	Navigation per Anker am Ende des Seiteninhalts	260
9.3.5	Toggle-Menü	264
9.3.6	Off-Canvas-Menü	271
9.4	Multilevel-Menüs	275
9.4.1	Die native Einbindung von Multilevel-Menüs auf iOS und Android ...	276
9.4.2	Andere Lösungen für Multilevel-Menüs	279
9.5	Zusammenfassung	282

10 Flexible Bildelemente	283
10.1 Anpassungsfähige Bilder	283
10.1.1 Praxisbeispiel: anpassungsfähiges Headerbild	285
10.1.2 Bilder ausschnittsweise anzeigen	287
10.1.3 Praxisbeispiel: nur Bildausschnitt für die Basisversion	287
10.1.4 Flexible Bilder, die nicht über die ganze Rasterbreite gehen	289
10.1.5 Praxisbeispiel: flexible Teaser-Boxen mit Bild und Text	289
10.2 Responsive Hintergrundbilder	293
10.2.1 Gekachelte Hintergrundmuster	293
10.2.2 Großflächige Hintergrundbilder	298
10.2.3 Praxisbeispiel: vollflächige Hintergrundbilder	302
10.2.4 Hintergrundgrafiken als Icons	305
10.2.5 CSS-Sprites für Hintergrundbilder	305
10.2.6 Praxisbeispiel: CSS-Sprites für hochauflösende Hintergrundgrafiken	307
10.3 Auflösungsunabhängige Grafiken (SVG)	311
10.3.1 Das Scalable-Vector-Graphics-Format	312
10.3.2 Praxisbeispiel: Icons als SVG-Sprite	312
10.3.3 Praxisbeispiel: SVG-Infografik versus GIF-Infografik	314
10.3.4 Einbindung von SVG-Grafiken	315
10.3.5 Praxisbeispiel: responsive SVG-Einbindung	321
10.3.6 Die responsive SVG-Einbindung	325
10.3.7 Browsersupport und Fallback	326
10.4 Icon-Fonts	330
10.4.1 Praxisbeispiel: Icon-Fonts mit Font Awesome	334
10.4.2 Icon-Fonts mit Ligaturen	336
10.5 Bilder für unterschiedliche Auflösungen	338
10.5.1 Wie ordne ich die richtigen Bilder im HTML-Code zu?	339
10.5.2 Beschnitt mit automatischer Berücksichtigung des Bildschwerpunkts	346
10.5.3 Und wer erstellt die ganzen Bilder?	348
10.6 Zusammenfassung	352

11	Mehr flexible Inhalte	355
11.1	Responsive Bildergalerien	356
11.1.1	Praxisbeispiel: flexible Slideshow mit ResponsiveSlides	356
11.1.2	Praxisbeispiel: responsive Bildergalerien mit Flexslider	358
11.1.3	Andere Bildergalerietools	364
11.2	Responsive Lightboxen	365
11.3	Responsive Image Maps	368
11.3.1	Praxisbeispiel: jQuery-rwdImageMaps.js	368
11.4	Anpassungsfähige Videos	370
11.4.1	Praxisbeispiel: HTML5-Videos	372
11.4.2	Responsive Embedding von YouTube & Co: Videos im iFrame	373
11.4.3	Automatische Anpassung für unterschiedliche Video-Seitenverhältnisse durch FitVids.js und FluidVids.js	377
11.4.4	HTML5-Videoplayer mit video.js	378
11.5	Adobe Flash	380
11.6	Flexible Karteneinbindungen (Google Maps)	381
11.7	Flexible Tabellen	383
11.7.1	Tabelleninhalte deaktivieren (hide on mobile)	384
11.7.2	Die Tabelle mit CSS umstrukturieren	386
11.7.3	Anpassungsfähige Tabellen mit FooTable	390
11.7.4	Mehr responsive Tabellen	392
11.8	Akkordeons und Inhaltsboxen mit Reitern	394
11.9	Flexible Formulare	398
11.9.1	Anwendungsbeispiel: Formular	399
11.10	Inhalte selektiv anzeigen und laden	404
11.10.1	Inhalte entfernen oder ergänzen – wann und wie?	404
11.10.2	Inhalte per CSS ausblenden (display: none)	405
11.10.3	Inhalte per CSS hinzufügen	406
11.10.4	Conditional Loading Content via JavaScript (und CSS)	408
11.11	Flexible Werbung	412
11.11.1	Bewusstsein schaffen	412
11.11.2	Alte und neue Bannerkonzepte	414
11.11.3	Bannererstellung und -auslieferung	416
11.11.4	Fixe Spalte und nur ein Rectangle-Format	418
11.11.5	Gezieltes Laden von Bannergrößenformaten mit Lazy-Ads	419
11.11.6	ZURB-Playground: Responsive Ads	421

11.11.7	Google AdSense Ads im responsiven Design ausliefern	422
11.11.8	Werbeblöcke zwischen Inhalte schieben mit CSS-Regions (Polyfill)	422
11.11.9	Inhalte dynamisch verschieben	424
11.12	Responsive HTML-E-Mails	426
11.13	Zusammenfassung	430

12 Qualitätssicherung und Optimierung 431

12.1	Testing	431
12.1.1	Validatoren für HTML und CSS	431
12.1.2	Breakpoints testen	432
12.1.3	Auf dem Gerät testen	440
12.1.4	Adobe Edge Inspect, SDKs	441
12.2	Performance testen und Optimierung	443
12.2.1	Das Performance-Budget	443
12.2.2	Was beeinträchtigt die Performance?	444
12.2.3	Skripte und HTML	447
12.2.4	Caching	457
12.2.5	Performanceoptimierung für Grafiken und Bilder	458
12.2.6	Schriften optimieren	460
12.2.7	Lazy Loading	463
12.2.8	Aus den Augen, aus dem Sinn?	468
12.3	Zusammenfassung	470

13 Fazit 471

Anhang 473

A.1	Website zum Buch	473
A.2	Quellennachweise der verwendeten Bilder	473

Index	477
-------------	-----