

Auf einen Blick

1	HTML	33
2	Hello Welt	49
3	Zufall, Spaß und Logik	85
4	CodeBreaker	111
5	Bubbles, Blasen und Arrays	137
6	Quiz	169
7	Rechenkönig	201
8	Textadventure	233
9	Hammurabi	269
10	Charts und Bibliotheken	305
11	Mondlandung	331
12	Im Tal der fallenden Steine	353
13	Objekte, Orakel, Schiffe und Seeungeheuer	367
14	Würfel, Torus, Raumschiffe	395
15	Zahlen, Buchstaben und KI	413

Inhalt

Vorwort	17
Materialien zum Buch	18
Über dieses Buch	19
1 HTML	33
<hr/>	
Die Befehle – Tags	34
Der Schlüssel zum Erfolg	35
(K)ein feierlicher Rahmen	36
Bitte nicht beachten!	38
Bevor es losgeht – eine Übersicht	40
Deine erste Webseite	41
Wie gehst du vor, um deine erste Seite zu erstellen?	42
Was kann HTML – und was kann es nicht?	44
Tags tieferlegen und verchromen – Attribute	45
Heiße Schmuggelware CSS	46
Und nicht vergessen: ändern, speichern, neu laden	47
2 Hallo Welt	49
<hr/>	
So kommen Programme in die Webseite	50
Ein kleines Meldungsfenster – der Einstieg	52
Und so kommt das JavaScript in die Webseite	53
Von Leerzeichen und Leerzeilen	54
Nur noch speichern und im Browser öffnen	55
Da geht noch was – ändern, speichern, neu laden	56
Das Handy fällt nicht weit vom Stamm – der gute alte Galileo und warum auch Formeln Spaß machen	57
Wie schreibe ich »V« in JavaScript?	58

Wohin mit dem Ergebnis?	59
Das alte Problem der Vergesslichkeit	61
Allheilmittel gegen Vergesslichkeit – die Variablen	63
Wie gut, dass es Variablen gibt	63
So speicherst du einen Wert in einer Variablen	64
Ein paar Regeln für Variablen	66
Falsch. FALSCH! FAAALSCH!!!	66
Richtige Variablennamen	67
Vom Handyweitwurf zum ersten Spiel	68
Mit der Formel kommt der Spaß	69
Jetzt schreiben wir das in JavaScript	69
Da stimmt etwas nicht: Wie aus Grad das Bogenmaß wird und warum	70
Des Pudels Kern – die eigentliche Berechnung	71
Lasst die Spiele beginnen	74
Eine Idee? Eine Idee – ein Szenario!	74
Zufall kann auch schön sein	74
Drei Versuche sollt ihr sein!	74
Die erste Schleife geht auch ohne Schnürsenkel	75
Ein detaillierter Blick in unsere Schleife	76
Waren da hinten nicht gerade noch der Bernd und die Alfi?	
Dort, wo jetzt das Monster steht?	78
Wenn schon, denn schon – Vergleiche mit »if«	79
Was du sonst noch machen kannst	80
Falls es mal nicht klappt	81

3 Zufall, Spaß und Logik

Zahlen raten	86
Die erste Überlegung – ganz einfach ausgedrückt	86
Ein neuer Versuch – ein klein wenig genauer	86
Was haben wir hier gemacht? Ein Programm geschrieben!	86
Die Anleitung haben wir – bauen wir unser Programm	87
Von der Beschreibung zum Programm	88

Was macht der Programmcode denn da?	89
Jetzt soll uns der Computer nach einer Zahl fragen	90
Zu groß, zu klein – wie wäre es mit einem kleinen Tipp?	90
Ja, wurde denn richtig geraten?	91
»Hey, mach's noch mal« – Schleifen mit »do-while«	92
Die »do-while«-Schleife	93
Über Bedingungen – größer, kleiner – und über das ominöse »!=«	94
Das fertige Programm	94
Übrigens, es gibt auch immer einen ganz anderen Weg	96
Schere, Stein, Papier	98
Computer schummeln nicht	99
Die Variablen festlegen	99
Was können wir daraus in JavaScript machen?	100
Jetzt in aller Ruhe	102
»else« – keine schöne Maid, eher ein »ansonsten« für alle (anderen) Fälle	105
Das »if« und das »else«	105
Sag mal, stimmt die Formel so? Formeln, Bauchgefühle, Tests	107

4 CodeBreaker

111

Die geheime Zahl	112
Von der Beschreibung zum Programm	113
Zahlen spalten einfach gemacht	115
Einfache Lösung mit Hausmitteln	115
Wie erfolgreich war das Raten?	117
Nur der Teufel steckt so tief im Detail und alle, die programmieren – »else if«	118
»else if« – ein starkes »ansonsten« mit einer weiteren Bedingung	119
Was jetzt noch fehlt – die anderen Zahlen, eine Ausgabe und 'ne tolle Schleife	120
Dann wollen wir mal den Rest machen	122
Tunen, tieferlegen, lackieren und Locken eindrehen	124
Zähl die gespielten Runden und begrenze das Spiel auf eine festgelegte Anzahl von Runden	125
Wenn's dann doch mal reicht – das Spiel selbst beenden	127

Nicht vergessen – wie war denn jetzt der Code?	129
Ein paar Zeilen als Einleitung	129
JavaScript über Klicks auf HTML-Elemente aufrufen	129
Funktionen – besonders dick und saugfähig	132
Und jetzt alles	134

5 Bubbles, Blasen und Arrays

137

Erst einmal alles fürs Sortieren	138
Arrays – die Vereinsmeier unter den Variablen	138
Werte lesen, schreiben und auch wieder vergessen	140
Einen Wert ändern	140
The sort must go on ... oder so ähnlich	142
Bubblesort – der nicht so ganz heilige Gral der abstrakten Rechenvorgänge	142
Bubblesort ohne Computer	143
Bubblesort mit Computer	144
Ready to rumble	145
Jetzt gibt's was in die Schleife	146
Die Sache mit »true« und »false«	147
Ein Durchgang macht noch keine fertige Liste	148
Eine Ausgabe muss her!	149
Alle Teile des Puzzles – unsortiert	149
Das fertige Puzzle	150
Feinschliff	151
Als Erstes machen wir eine Funktion aus unserem Bubblesort	152
Zwei Listen sollt ihr sein	152
Mehr als nur Feinheiten – du und deine Funktion	153
Schön und auch noch zeitgesteuert	156
Das Ende der weißen Seiten ist nahe	157
Wie sieht das für die Ausgabe unseres Bubblesort aus?	158
HTML – das vernachlässigte Stiefkind der Aktualisierung	159
Erst einmal das Handwerkszeug – zeitgesteuerte Aufrufe	160
Richtiges temporales Zaubern für Anfänger	161
Etwas schicke Kosmetik	165
Die volle Funktion für Bubblesort	165

6 Quiz

169

Tieferlegen und verchromen – alles mit CSS	175
Dreimal darfst du raten	178
Passend zum Quiz: Rate die Variablen	179
Auch ganz passend: Rate die Funktionen	181
Fragen, Antworten und die richtige Lösung – wohin damit?	183
Vom richtigen Mischen und von anonymen Funktionen	185
Nur leicht geschüttelt, nicht gerührt	187
Die Sache mit der Henne, dem Ei und dem besten Anfang	188
Also gut, zuerst die Funktion »tippeButton«	189
Schönheit löst keine Probleme – ist aber ganz schön!	191
Einmal Rot bitte – falsche Antwort	191
Das Quiz starten	193
Die gedrückten Buttons nicht vergessen	195

7 Rechenkönig

201

Die Benutzeroberfläche	202
Zuerst die Funktionen und die Variablen	206
Was ist zu tun in der Funktion »stelleAufgabe«?	211
Zwei Zahlen sollt ihr sein	212
Die bessere Lösung – »switch-case«	213
»switch«-»case« für unser Programm	215
Keine negativen Ergebnisse	216
Der Spieler ist am Zug	218
Der Name wird Programm: »pruefeEingabe«	218
Stimmt das Ergebnis?	219
Das Programm als Ganzes	220
Nicht für die Ewigkeit – aber länger als nur für eine Sitzung	223
Auch das Laden will gelernt sein	225
Holen wir unsere Zahlen – als echte Zahlen	227
Und sogar das Löschen will gelernt sein	228
Was fehlt noch? Ist noch etwas zu tun?	232

8 Textadventure

233

Wie setzen wir das um?	235
JSON – ein kuscheliges Zuhause für Daten und Geschichten	238
Eine Passage macht noch keine Geschichte	239
Nicht nur Türen brauchen einen Schlüssel	240
Zeit für etwas HTML und CSS	242
Von JSON zu JavaScript	246
Die objektorientierte Notation	247
Zuerst die grundlegende Funktionalität – der Prototyp	250
Nach dem Prototyp	254
Aus den Nummern die wirklichen Texte holen	256
Was muss die neue Funktion tun?	257
Teile und herrsche – mehr als nur eine Datei	262
Die Datei »monitor.css«	263
Die Datei »abenteuerJson.js«	264
Die Datei »abenteuer.js«	266
Zu guter Letzt – unser HTML in der »abenteuer.html«	267

9 Hammurabi

269

Wie funktioniert das Spiel?	270
Ein wenig HTML	271
Und noch eine Portion CSS	272
Die Regeln – im Detail	274
Lass die Spiele beginnen	279
Ein Bericht für den Herrscher – die Ausgabe	281
Der grundlegende Text wird gebastelt	281
Unsere Zufallszahlen	286
Eine Spielrunde – ein ganzes Jahr	288
Die Eingabe – dem Volk Befehle erteilen	290
Mahlzeit und Prost – wir verteilen Nahrungsmittel	293

Die Aussaat	294
Zu guter Letzt noch etwas Handel	295
Das Ende ist näher, als du denkst	297
Das ganze Programm in einem Rutsch	299

10 Charts und Bibliotheken 305

Chartis.js	307
Woher nehmen und nicht stehlen?	308
Wie funktioniert es?	308
So sieht es mit CDN aus	312
Gestatten? Daten, Daten im JSON-Format	314
Frei wählbar, die Optionen	314
Der eigentliche Star und Hauptdarsteller: Das Objekt	315
Zeit für eigene Daten	316
Mit »undefined« ist schlecht zählen	319
Noch schnell die Labels – die Beschriftung der x-Achse	320
Zeit für Änderungen	322
Eine zweite Zufallszahl soll es sein	324

11 Mondlandung 331

Was brauchen wir auf unserer Webseite?	332
Schöner abstürzen	334
Ein paar Funktionen wären auch ganz hilfreich	334
Auch das schönste Programm ist nichts ohne eine Ausgabe	338
Etwas Kontrolle muss sein	341
Schöner fallen mit Canvas und JavaScript	344
Mehr Farbe im Leben und auf der Planetenoberfläche	344
Canvas im JavaScript	346

12 Im Tal der fallenden Steine

353

Die HTML-Datei	354
Der Code	355
Kein Programm ist so schwer wie das, das du nicht selbst geschrieben hast	365

13 Objekte, Orakel, Schiffe und Seeungeheuer

367

Klassen, Objekte und die alten Griechen	368
Ein Orakel und die erste eigene Klasse	369
Die Klasse	370
Ein paar Attribute brauchen wir schon mal	371
Das hat Methode(n)	371
Eine zweite Methode – gib uns ein Element	373
Die Klasse haben wir – Zeit für ein Objekt	375
Das erste eigene Objekt	375
Noch eine Schippe OOP obendrauf	377
Einmal das volle Programm, bitte	378
Setzt die Segel!	379
So schreiben wir unser Programm	380
Eine Karte für die hohe See	382
Einmal die Karte, bitte	386
Das Schiff	387
Eine Steuerung – ganz klassisch	389
Die Klasse »Karte«	391
Die Klasse »Schiff«	393
Und noch die Steuerung	394

14 Würfel, Torus, Raumschiffe

395

Three.js – woher nehmen und nicht stehlen?	396
Das erste Mal in 3D – fast wie am Filmset	398
Hereinspaziert und willkommen in deinem eigenen Filmstudio!	398
Das Filmset wird festgelegt	398
Die Kamera wird aufgestellt	399
Die Beleuchtung muss stimmen	399
Der Hauptdarsteller betritt das Set	400
Film ab	401
Nur noch ein paar Regieanweisungen	401
Szene Reloaded	402
Alles eine Frage der Perspektive	403
Und hier der Code am Stück	404
Ein Raumschiff – ganz klassisch	406

15 Zahlen, Buchstaben und KI

413

Was fange ich mit künstlicher Intelligenz an?	414
Zeit für einen Test	419
Die Frage	419
Die Antwort	419
Diesmal etwas schwerer	421
Und jetzt noch einmal alles am Stück	421
Mehr KI und JavaScript	425
 Index	427