

# Inhalt

Vorwort .....	15
Materialien zum Buch .....	16
Über dieses Buch .....	17

---

## 1 HTML 31

---

<b>Die Befehle: Tags .....</b>	32
Der Schlüssel zum Erfolg .....	33
(K)ein feierlicher Rahmen .....	34
Bitte nicht beachten! .....	36
Bevor es losgeht – eine Übersicht .....	38
<b>Deine erste Webseite .....</b>	39
Wie gehst du vor, um deine erste Seite zu erstellen? .....	40
Was kann HTML – und was kann es nicht? .....	42
<b>Tags tieferlegen und verchromen: Attribute .....</b>	43
Heiße Schmuggelware CSS .....	44
Und nicht vergessen: ändern, speichern, neu laden .....	45

---

## 2 Hallo Welt 47

---

<b>So kommen Programme in die Webseite .....</b>	48
<b>Ein kleines Meldungsfenster – der Einstieg .....</b>	50
<b>Und so kommt das JavaScript in die Webseite .....</b>	51
Von Leerzeichen und Leerzeilen .....	52
Nur noch speichern und im Browser öffnen .....	53
Da geht noch was – ändern, speichern, neu laden .....	54
<b>Das Handy fällt nicht weit vom Stamm – der gute alte Galileo und warum auch Formeln Spaß machen .....</b>	55
<b>Wie schreibe ich »V« in JavaScript? .....</b>	56

<b>Wohin mit dem Ergebnis? .....</b>	57
Das alte Problem der Vergesslichkeit .....	60
<b>Allheilmittel gegen Vergesslichkeit – die Variablen .....</b>	61
Wie gut, dass es Variablen gibt .....	61
So speicherst du einen Wert in einer Variablen .....	62
Ein paar Regeln für Variablen .....	64
Falsch. FALSCH! FAAALSCH!!! .....	65
Richtige Variablennamen .....	65
<b>Von Handyweitwurf zum ersten Spiel .....</b>	67
Mit der Formel kommt der Spaß .....	67
Jetzt schreiben wir das in JavaScript .....	67
Da stimmt etwas nicht: Wie aus Grad das Bogenmaß wird – und warum .....	68
Des Pudels Kern – die eigentliche Berechnung .....	69
<b>Lasst die Spiele beginnen .....</b>	72
Eine Idee? Eine Idee – ein Szenario! .....	72
Zufall kann auch schön sein .....	72
Drei Versuche sollt ihr sein! .....	72
<b>Die erste Schleife geht auch ohne Schnürsenkel .....</b>	73
Ein detaillierter Blick in unsere Schleife .....	74
<b>Waren da hinten nicht gerade noch der Bernd und die Alfi?</b>	
<b>Dort, wo jetzt das Monster steht? .....</b>	76
Wenn schon, denn schon – Vergleiche mit »if« .....	77
Was du sonst noch machen kannst .....	78
<b>Falls es mal nicht klappt .....</b>	79

---

<b>3 Zufall, Spaß und Logik</b>	81
<b>Zahlenraten .....</b>	82
Die erste Überlegung – ganz einfach ausgedrückt .....	82
Ein neuer Versuch – ein klein wenig genauer .....	82
Was haben wir hier gemacht? Ein Programm geschrieben! .....	82
<b>Die Anleitung haben wir – bauen wir unser Programm .....</b>	83
Von der Beschreibung zum Programm .....	84

<b>Was macht der Programmcode denn da? .....</b>	85
Jetzt soll uns der Computer nach einer Zahl fragen .....	86
Zu groß, zu klein – wie wäre es mit einem kleinen Tipp? .....	86
Ja, wurde denn richtig geraten? .....	87
<b>»Hey, mach's noch mal« – Schleifen mit »do-while« .....</b>	88
Die »do-while«-Schleife .....	89
Über Bedingungen: Größer, kleiner – und das ominöse != .....	90
<b>Das fertige Programm .....</b>	90
Übrigens, es gibt auch immer einen ganz anderen Weg .....	92
<b>Schere, Stein, Papier .....</b>	94
Computer schummeln nicht .....	95
Die Variablen festlegen .....	95
Was können wir daraus in JavaScript machen? .....	95
Jetzt in aller Ruhe .....	98
»else« – keine schöne Maid, eher ein »ansonsten« für alle (anderen) Fälle .....	101
<b>Das »if« und das »else« .....</b>	101
<b>Sag mal, stimmt die Formel so? Formeln, Bauchgefühle, Tests .....</b>	102

---

## 4 CodeBreaker 105

<b>Die geheime Zahl .....</b>	106
<b>Von der Beschreibung zum Programm .....</b>	107
Zahlen spalten einfach gemacht .....	109
Einfache Lösung mit Hausmitteln .....	109
Wie erfolgreich war das Raten? .....	111
<b>Nur der Teufel steckt so tief im Detail – und Programmierer: »else if« .....</b>	112
»else if« – ein starkes »ansonsten« mit einer weiteren Bedingung .....	112
Was jetzt noch fehlt – die anderen Zahlen, eine Ausgabe und 'ne tolle Schleife .....	114
Dann wollen wir mal den Rest machen .....	116
<b>Tunen, tieferlegen, lackieren und Locken eindrehen .....</b>	118
Zähl die gespielten Runden, und begrenze das Spiel auf eine festgelegte Anzahl von Runden .....	119
Wenn's dann doch mal reicht – das Spiel selbst beenden .....	121

Nicht vergessen – wie war denn jetzt der Code? .....	122
Ein paar Zeilen als Einleitung .....	122
<b>JavaScript über Klicks auf HTML-Elemente aufrufen .....</b>	<b>122</b>
Funktionen – besonders dick und saugfähig .....	125
Und jetzt alles .....	128

---

## 5 Bubbles, Blasen und Arrays 133

---

<b>Erst einmal alles fürs Sortieren .....</b>	<b>134</b>
<b>Arrays – die Vereinsmeier unter den Variablen .....</b>	<b>134</b>
<b>Werte lesen, schreiben und auch wieder vergessen .....</b>	<b>136</b>
Einen Wert ändern .....	136
<b>The sort must go on ... oder so ähnlich .....</b>	<b>138</b>
Bubblesort – der nicht so ganz heilige Gral der abstrakten Rechenvorgänge .....	138
Bubblesort ohne Computer .....	139
<b>Bubblesort mit Computer .....</b>	<b>140</b>
Ready to rumble .....	141
Jetzt gibt's was in die Schleife .....	141
Die Sache mit »true« und »false« .....	143
Ein Durchgang macht noch keine fertige Liste .....	144
Eine Ausgabe muss her! .....	145
Alle Teile des Puzzles – unsortiert .....	145
Das fertige Puzzle .....	146
<b>Feinschliff .....</b>	<b>147</b>
Als Erstes: Wir machen eine Funktion aus unserem Bubblesort .....	147
Zwei Listen sollt ihr sein .....	148
Mehr als nur Feinheiten – du und deine Funktion .....	149
Schön und auch noch zeitgesteuert .....	152
<b>Das Ende der weißen Seiten ist nahe .....</b>	<b>152</b>
Wie sieht das für die Ausgabe von unserem Bubblesort aus? .....	153
HTML – das vernachlässigte Stiefkind der Aktualisierung .....	155
Erst einmal das Handwerkszeug – zeitgesteuerte Aufrufe .....	155
Richtiges temporales Zaubern für Anfänger .....	156
<b>Etwas schicke Kosmetik .....</b>	<b>160</b>
<b>Die volle Funktion für Bubblesort .....</b>	<b>161</b>

<b>Tieferlegen und verchromen – alles mit CSS .....</b>	169
<b>Dreimal darfst du raten .....</b>	172
Passend zum Quiz: Rate die Variablen .....	173
Auch ganz passend: Rate die Funktionen .....	175
Fragen, Antworten und die richtige Lösung. Wohin damit? .....	177
<b>Vom richtigen Mischen und von anonymen Funktionen .....</b>	179
Nur leicht geschüttelt – nicht gerührt .....	181
<b>Die Sache mit der Henne, dem Ei und dem besten Anfang .....</b>	182
Also gut, zuerst die Funktion »tippeButton« .....	183
Schönheit löst keine Probleme – ist aber ganz schön! .....	185
Einmal Rot bitte – falsche Antwort .....	186
<b>Das Quiz starten .....</b>	187
Nicht vergessen – die gedrückten Buttons .....	189

<b>Die Benutzeroberfläche .....</b>	196
<b>Zuerst die Funktionen und die Variablen .....</b>	200
Was ist in der Funktion »stelleAufgabe« zu tun? .....	205
Zwei Zahlen sollt ihr sein .....	206
Und zwar »switch«->»case« .....	207
»switch«->»case« für unser Programm .....	209
Keine negativen Ergebnisse .....	210
<b>Der Spieler ist am Zug .....</b>	212
Der Name wird Programm: »pruefeEingabe« .....	212
Stimmt das Ergebnis? .....	212
<b>Das Programm als Ganzes .....</b>	214
<b>Nicht für die Ewigkeit – aber länger als nur für eine Sitzung .....</b>	216
Auch das Laden will gelernt sein .....	219
Holen wir unsere Zahlen – als echte Zahlen .....	221
Und sogar das Löschen will gelernt sein .....	222
Was fehlt noch? Ist noch etwas zu tun? .....	225

## 8 Textadventure

227

---

Wie setzen wir das um? .....	229
JSON – ein kuscheliges Zuhause für Daten und Geschichten .....	232
Eine Passage macht noch keine Geschichte .....	233
Nicht nur Türen brauchen einen Schlüssel .....	234
Zeit für etwas HTML und CSS .....	236
Von JSON zu JavaScript .....	240
Die objektorientierte Notation .....	240
Zuerst die grundlegende Funktionalität – der Prototyp .....	244
Nach dem Prototyp .....	248
Aus den Nummern die wirklichen Texte holen .....	250
Was muss die neue Funktion tun? .....	251
Teile und herrsche – mehr als nur eine Datei .....	256
Die Datei »monitor.css« .....	257
Die Datei »abenteuerJson.js« .....	258
Die Datei »abenteuer.js« .....	259
Zu guter Letzt – unser HTML in der »abenteuer.html« .....	260

## 9 Hammurabi

263

---

Wie funktioniert das Spiel? .....	264
Ein wenig HTML .....	265
Und noch eine Portion CSS .....	266
Die Regeln – im Detail .....	269
Lass die Spiele beginnen .....	274
Ein Bericht für den Herrscher – die Ausgabe .....	275
Der grundlegende Text wird gebastelt .....	276
Unsere Zufallszahlen .....	280
Eine Spielrunde – ein ganzes Jahr .....	283
Die Eingabe – dem Volk Befehle erteilen .....	285
Mahlzeit und Prost – wir verteilen Nahrungsmittel .....	288

Die Aussaat .....	289
Zu guter Letzt noch etwas Handel .....	290
Das Ende ist näher, als du denkst .....	292
<b>Das ganze Programm in einem Rutsch .....</b>	<b>293</b>

---

## 10 Charts und Bibliotheken

299

<b>Chartist.js .....</b>	<b>301</b>
Woher nehmen und nicht stehlen? .....	302
Wie funktioniert es? .....	302
Gestatten? Daten, Daten im JSON-Format .....	307
Frei wählbar, die Optionen .....	308
Der eigentliche Star und Hauptdarsteller: Das Objekt .....	308
<b>Zeit für eigene Daten .....</b>	<b>310</b>
Mit »undefined« ist schlecht zählen .....	313
Noch schnell die Labels – die Beschriftung der X-Achse .....	314
Zeit für Änderungen .....	316
Eine zweite Zufallszahl soll es sein .....	318

---

## 11 Mondlandung

325

<b>Was brauchen wir auf unserer Webseite? .....</b>	<b>326</b>
<b>Schöner abstürzen .....</b>	<b>328</b>
<b>Ein paar Funktionen wären auch ganz hilfreich .....</b>	<b>328</b>
<b>Das schönste Programm ist nichts ohne eine Ausgabe .....</b>	<b>332</b>
<b>Etwas Kontrolle muss sein .....</b>	<b>335</b>
<b>Schöner fallen mit Canvas und JavaScript .....</b>	<b>338</b>
Mehr Farbe im Leben und auf der Planetenoberfläche .....	338
Canvas im JavaScript .....	340

## 12 Im Tal der fallenden Steine

347

---

Die HTML-Datei .....	348
Der Code .....	349
Kein Programm ist so schwer wie das, das du nicht selbst geschrieben hast .....	359

## 13 Objekte, Orakel, Schiffe und Seeungeheuer

361

---

Klassen, Objekte und die alten Griechen .....	362
Ein Orakel und die erste eigene Klasse .....	363
Die Klasse .....	364
Ein paar Attribute brauchen wir schon mal .....	365
Das hat Methode(n) .....	365
Eine zweite Methode: Gib uns ein Element .....	367
Die Klasse haben wir – Zeit für ein Objekt .....	369
Das erste eigene Objekt .....	369
Noch eine Schippe OOP oben drauf .....	371
Einmal das volle Programm, bitte .....	372
Setzt die Segel! .....	373
So schreiben wir unser Programm .....	374
Eine Karte für die hohe See .....	376
Einmal die Karte, bitte .....	380
Das Schiff .....	381
Eine Steuerung – ganz klassisch .....	383
Die Klasse »Karte« .....	385
Die Klasse »Schiff« .....	386
Und noch die Steuerung .....	387
Index .....	389