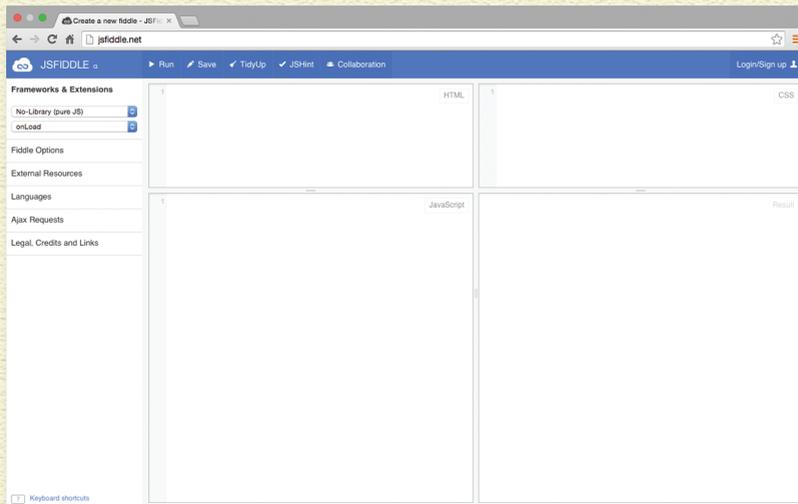


KAPITEL 1 AUF DIE PLÄTZE



OHNE FREMDE HILFE KÖNNEN COMPUTER GAR NICHTS. Sie brauchen ein Computerprogramm, das ihnen sagt, was sie tun sollen, und diese Computerprogramme müssen von Menschen geschrieben werden. Die Leute, die den Code schreiben, der dafür sorgt, dass Computer alles Mögliche machen, nennt man *Programmierer*.

WAS IST PROGRAMMIEREN?

Ein **Computerprogramm** ist eine Zusammenstellung von Anweisungen, die ein Computer verstehen und befolgen kann. Ein anderer Name für Computerprogramme ist *Software*. Diese Anweisungen zu schreiben, nennt man *Programmieren* oder auch *Codieren*.

```
1 <div id="robot">
2 <div id="head">
3 <div class="eye" id="righteye"></div>
4 <div class="eye" id="lefteye"></div>
5 <div id="nose"></div>
6 <div id="mouth"></div>
7 </div>
8 <div class="arm" id="rightarm"></div>
9 <div id="body"><p id="message">I Love to Code!</p></div>
10
11 var rightEye = document.getElementById("righteye");
12 var leftEye = document.getElementById("lefteye");
13 var leftArm = document.getElementById("leftarm");
14
15 rightEye.addEventListener("click", moveUpDown);
16 leftEye.addEventListener("click", moveUpDown);
17 leftArm.addEventListener("click", moveRightLeft);
18
19
20 function moveUpDown(e) {
21   var robotPart = e.target;
22   var top = 0;
23
24   var id = setInterval(frame, 10) // draw every 10ms
25
26   function frame() {
27     robotPart.style.top = top + '%';
28     top++;
29     if (top === 20){
30       clearInterval(id);
31     }
32   }
33 }
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54
55
56
57
58
59
60
61
62
63
64
65
66
67
68
69
70
71
72
73
74
75
76
77
78
79
80
81
82
83
84
85
86
87
88
89
90
91
92
93
94
95
96
97
98
99
100
101
102
103
104
105
106
107
108
109
110
111
112
113
114
115
116
117
118
119
120
121
122
123
124
125
126
127
128
129
130
131
132
133
134
135
136
137
138
139
140
141
142
143
144
145
146
147
148
149
150
151
152
153
154
155
156
157
158
159
160
161
162
163
164
165
166
167
168
169
170
171
172
173
174
175
176
177
178
179
180
181
182
183
184
185
186
187
188
189
190
191
192
193
194
195
196
197
198
199
200
201
202
203
204
205
206
207
208
209
210
211
212
213
214
215
216
217
218
219
220
221
222
223
224
225
226
227
228
229
230
231
232
233
234
235
236
237
238
239
240
241
242
243
244
245
246
247
248
249
250
251
252
253
254
255
256
257
258
259
260
261
262
263
264
265
266
267
268
269
270
271
272
273
274
275
276
277
278
279
280
281
282
283
284
285
286
287
288
289
290
291
292
293
294
295
296
297
298
299
300
301
302
303
304
305
306
307
308
309
310
311
312
313
314
315
316
317
318
319
320
321
322
323
324
325
326
327
328
329
330
331
332
333
334
335
336
337
338
339
340
341
342
343
344
345
346
347
348
349
350
351
352
353
354
355
356
357
358
359
360
361
362
363
364
365
366
367
368
369
370
371
372
373
374
375
376
377
378
379
380
381
382
383
384
385
386
387
388
389
390
391
392
393
394
395
396
397
398
399
400
401
402
403
404
405
406
407
408
409
410
411
412
413
414
415
416
417
418
419
420
421
422
423
424
425
426
427
428
429
430
431
432
433
434
435
436
437
438
439
440
441
442
443
444
445
446
447
448
449
450
451
452
453
454
455
456
457
458
459
460
461
462
463
464
465
466
467
468
469
470
471
472
473
474
475
476
477
478
479
480
481
482
483
484
485
486
487
488
489
490
491
492
493
494
495
496
497
498
499
500
501
502
503
504
505
506
507
508
509
510
511
512
513
514
515
516
517
518
519
520
521
522
523
524
525
526
527
528
529
530
531
532
533
534
535
536
537
538
539
540
541
542
543
544
545
546
547
548
549
550
551
552
553
554
555
556
557
558
559
560
561
562
563
564
565
566
567
568
569
570
571
572
573
574
575
576
577
578
579
580
581
582
583
584
585
586
587
588
589
590
591
592
593
594
595
596
597
598
599
600
601
602
603
604
605
606
607
608
609
610
611
612
613
614
615
616
617
618
619
620
621
622
623
624
625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671
672
673
674
675
676
677
678
679
680
681
682
683
684
685
686
687
688
689
690
691
692
693
694
695
696
697
698
699
700
701
702
703
704
705
706
707
708
709
710
711
712
713
714
715
716
717
718
719
720
721
722
723
724
725
726
727
728
729
730
731
732
733
734
735
736
737
738
739
740
741
742
743
744
745
746
747
748
749
750
751
752
753
754
755
756
757
758
759
760
761
762
763
764
765
766
767
768
769
770
771
772
773
774
775
776
777
778
779
780
781
782
783
784
785
786
787
788
789
790
791
792
793
794
795
796
797
798
799
800
801
802
803
804
805
806
807
808
809
810
811
812
813
814
815
816
817
818
819
820
821
822
823
824
825
826
827
828
829
830
831
832
833
834
835
836
837
838
839
840
841
842
843
844
845
846
847
848
849
850
851
852
853
854
855
856
857
858
859
860
861
862
863
864
865
866
867
868
869
870
871
872
873
874
875
876
877
878
879
880
881
882
883
884
885
886
887
888
889
890
891
892
893
894
895
896
897
898
899
900
901
902
903
904
905
906
907
908
909
910
911
912
913
914
915
916
917
918
919
920
921
922
923
924
925
926
927
928
929
930
931
932
933
934
935
936
937
938
939
940
941
942
943
944
945
946
947
948
949
950
951
952
953
954
955
956
957
958
959
960
961
962
963
964
965
966
967
968
969
970
971
972
973
974
975
976
977
978
979
980
981
982
983
984
985
986
987
988
989
990
991
992
993
994
995
996
997
998
999
1000
```

Im Augenblick wird dir dieser Computercode verwirrend erscheinen, aber wenn du dieses Buch gelesen hast, wirst du nicht nur wissen, wie man ihn liest, sondern ihn auch selbst schreiben können!

Unsere heutigen Computer sind leistungsfähiger als je zuvor. Früher brauchte man einen ganzen Raum voller Computer, um die Arbeit zu erledigen, die heute schon ein Smartphone schafft. Eines haben allerdings alle Computer gemeinsam, egal wie leistungsfähig sie auch sind: Ohne Computerprogramme leisten sie gar nichts, und diese Computerprogramme müssen Programmierer erst einmal schreiben.



DIE FRAU, DIE DAS PROGRAMMIEREN ERFAND

Die ersten elektronischen Computer sind in den 1930er-Jahren entwickelt worden. Aber schon Mitte des 19. Jahrhunderts wurde das erste Computerprogramm – mehrere Anweisungen, die von einer Maschine ausgeführt werden sollten – geschrieben. Die Autorin dieses ersten Computerprogramms und damit die erste Programmiererin war eine Frau namens Ada Lovelace. Sie war Mathematikerin und lebte in England. Ada war die erste Person, die sich eine Welt vorstellen konnte, in der Computer all das können, wofür wir sie heute benutzen, zum Beispiel Texte schreiben, Bilder anzeigen und Musik abspielen.



Foto mit Genehmigung von Jitze Couperus
([https://commons.wikimedia.org/wiki/File:Babbage_Difference_Engine_\(Being_utilised\).jpg](https://commons.wikimedia.org/wiki/File:Babbage_Difference_Engine_(Being_utilised).jpg))

Der Computer, für den Ada Lovelace Programme schrieb, sah ganz anders aus als unsere heutigen Computer. Statt eines Steckers oder eines Akkus hatte er eine Kurbel! Schaut euch mal das Foto von Charles Babbages **Differenzmaschine** an, einem der ersten Computer überhaupt:

Um Computerprogramme für Maschinen verständlich zu machen, braucht man Compiler. *Compiler* sind Programme, die Programmiersprachen in Maschinensprache übersetzen. Den allerersten Compiler entwickelte Grace Hopper im Jahre 1944. Ihre Erfindung machte es möglich, Computerprogramme auf verschiedenen Computertypen laufen zu lassen. Grace Hopper hat auch den Begriff *Debugging* erfunden – so nennt man die Fehlersuche bei Computerprogrammen. *Bug* ist ein englisches Wort für Insekt. (Beim Ausprobieren eines Computerprogramms auf einem der frühen Computer fand Grace Hopper eine Motte, die irgendwie zwischen die Kontakte geraten war und die Ausführung des Programms verhinderte.)

Die Computer, mit denen Grace Hopper arbeitete, liefen mit Strom (nicht mit Kurbeln) und waren riesig. Das Foto zeigt **Colossus**, den ersten programmierbaren elektronischen Computer.

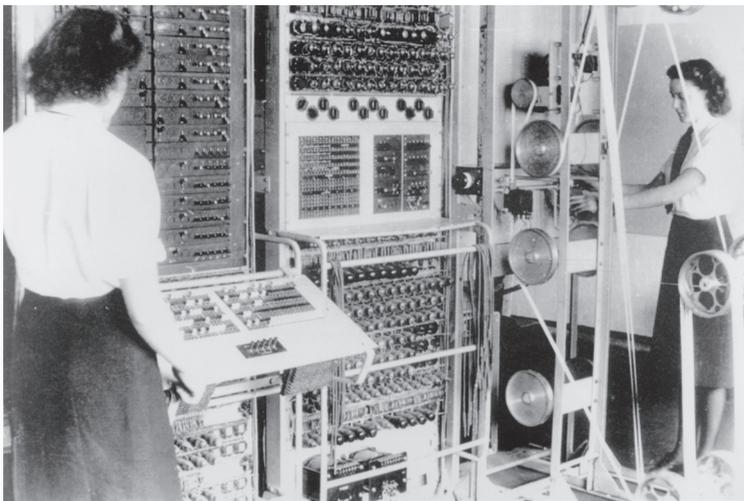


Foto mit Genehmigung des Bletchley Park Trust/Science & Society Picture Library (www.filckr.com/photos/101251639@N02/9669449367)

Computerprogramme helfen uns bei vielen alltäglichen Dingen. Unter anderem können sie:

- » Musik und Videos abspielen
- » Wissenschaftliche Experimente durchführen
- » Autos entwerfen
- » Maschinen erfinden
- » Spiele spielen
- » Roboter steuern
- » Satelliten und Raumschiffe steuern
- » Zeitungen und Magazine gestalten
- » Beim Lernen helfen

Fallen dir noch andere Sachen ein, die Computer können?

Am Anfang jedes Computerprogramms steht eine Idee. Bevor du die erste Codezeile schreibst, musst du erst einmal überlegen, was das Programm machen soll. Nutze deine Vorstellungskraft und träume munter drauflos!



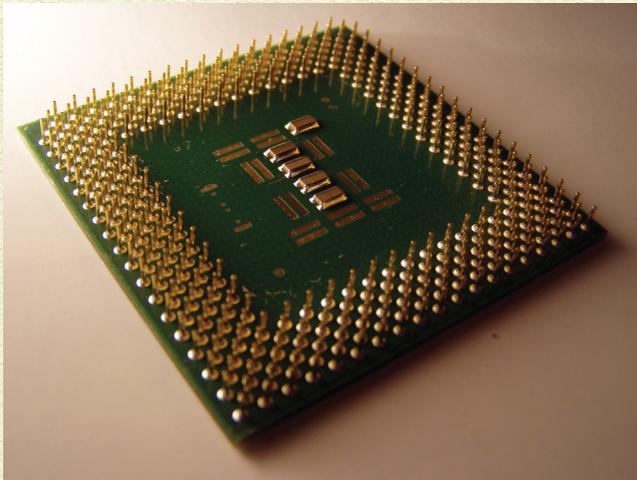
Hier sind ein paar hilfreiche Fragen, über die du nachdenken solltest, bevor du mit dem Programmieren anfängst:

- » Was wird mein Programm machen?
- » Wer wird mein Programm verwenden?
- » Warum werden sie mein Programm verwenden?
- » Wo werden sie mein Programm verwenden?
- » Wie werden sie mein Programm verwenden?

WIE KANN MAN MIT COMPUTERN SPRECHEN?

Jedes Computerprogramm wird in einer bestimmten Programmiersprache geschrieben. Programmiersprachen ermöglichen dir, Anweisungen zu schreiben, die in Maschinensprache übersetzt (oder kompiliert) werden können. Letztendlich werden Programmiersprachen in *binären Code* umgewandelt, der Buchstaben, Zahlen und Symbole nur mit den Ziffern »0« und »1« darstellt. Dazu gleich mehr.

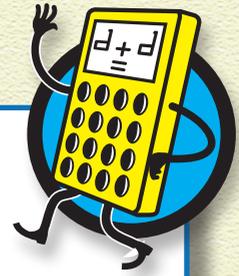
Das Herz jedes Computers ist der **Prozessor** (Central Processing Unit/CPU). Ein Prozessor besteht aus Millionen winziger, äußerst schneller Schalter (*Transistoren* genannt), die entweder an- oder ausgeschaltet sein können. Die Positionen dieser Schalter bestimmen, was der Computer macht. Die Software, die von Programmierern geschrieben wird, sagt den Schaltern mithilfe von binärem Code, wann sie an und wann sie aus sein sollen.



Quelle: <https://commons.wikimedia.org/wiki/File:Cpu.jpg>

Ausnahmslos alles, was ein Computer macht, ist das Ergebnis einer bestimmten Kombination vieler Nullen und Einsen. Um den Kleinbuchstaben »a« anzuzeigen, verwenden Computer zum Beispiel den folgenden binären Code:

0110 0001



WIE MAN ZAHLEN IM BINÄREN CODE SCHREIBT

Möchtest du lernen, wie ein Computer zu zählen?
Wir zeigen dir, wie du eine beliebige Zahl zwischen 0 und 255
binär darstellen kannst. Als Beispiel nehmen wir die Zahl 150.

- 1** Schreibe die folgenden Zahlen oben auf ein Blatt Papier
in eine Reihe:

128	64	32	16	8	4	2	1
-----	----	----	----	---	---	---	---

Diese Zahlen sind dein binärer Spickzettel.

- 2** Schreibe auf einen anderen Zettel oder irgendwo an
den Rand die Zahl, die du binär darstellen möchtest,
also die Zahl 150.
- 3** Beginne mit der ersten Zahl, die du auf dein Blatt
Papier geschrieben hast, 128. Wenn diese Zahl in 150
hineinpasst, schreibe die Ziffer 1 darunter. Passt sie
nicht in 150 hinein, schreibe die Ziffer 0 darunter.

Weil 128 in 150 hineinpasst, schreib die Ziffer 1 unter
die 128.

128	64	32	16	8	4	2	1
1							

- 4** Subtrahiere die Zahl 128 von der Zahl, die du in binärem
Code darstellen willst (150): $150 - 128 = 22$. Streiche die
Zahl 150 durch und schreibe stattdessen die Zahl 22 auf.

- 5** Vergleiche den Rest mit der nächsten Zahl auf deinem binären Spickzettel, 64. Passt 64 in 22? Nein, also schreibst du eine 0 unter die 64 und gehst zur nächsten Zahl auf deinem binären Spickzettel weiter, 32.
- 6** Weil 32 auch nicht in 22 passt, schreibst du eine 0 unter die 32 und gehst zur nächsten Zahl auf deinem binären Spickzettel – das ist die Zahl 16.
- 7** Aha! 16 passt in 22! Schreib eine 1 unter die 16 und subtrahiere dann 16 von 22: $22 - 16 = 6$.
- 8** Schau dir die nächste Zahl auf deinem binären Spickzettel an. Passt 8 in 6? Nein, also schreibst du unter die 8 eine 0 und gehst zur nächsten Zahl weiter.

So machst du weiter, bis du die letzte Zahl auf deinem binären Spickzettel erreicht hast. Am Ende sollte deine binäre Darstellung so aussehen:

128	64	32	16	8	4	2	1
1	0	0	1	0	1	1	0

Jetzt weißt du, dass 10010110 der binäre Code für die Zahl 150 ist.

Jede Null oder Eins in einer binären Zahl nennt man ein *Bit*. Eine Kombination aus acht Bits nennt man ein *Byte*. Wahrscheinlich kennst du die Wörter *Kilobyte*, *Megabyte* und *Gigabyte*, die im Zusammenhang mit der Größe von Speichern verwendet werden. Damit ist die Anzahl der Bytes gemeint, die in den Speicher passen.

Die Tabelle zeigt die **gebräuchlichsten Speichergrößen**.

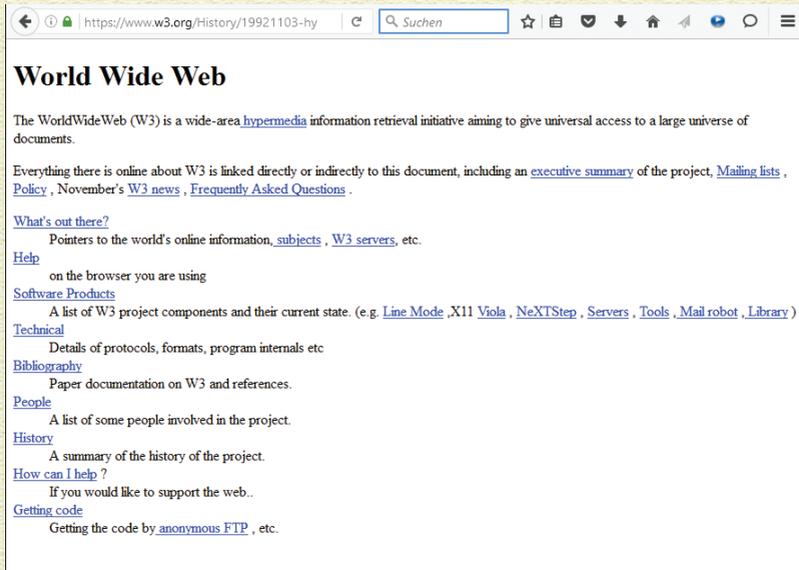
Wie viele Bytes sind das?		
<i>Name</i>	<i>Anzahl der Bytes</i>	<i>Was kann man speichern</i>
Kilobyte (KB)	1.024	Zwei oder drei Textabsätze
Megabyte (MB)	1.048.576	800 Buchseiten
Gigabyte (GB)	1.079.741.824	1.000 Bücher
Terabyte (TB)	1.099.511.627.776	Eine Million Bücher
Petabyte (PB)	1.125.899.906.842.624	Mehrere Ausgaben aller jemals geschriebenen Bücher!

Ein typisches kleines Computerprogramm kann zwischen ein paar Kilobytes und ein paar Megabytes an Anweisungen, Bildern und anderen Daten enthalten. Wir alle haben nicht genug Zeit, Tausende Nullen und Einsen für ein Computerprogramm zu tippen. Weil Computer nicht verstehen, was wir meinen, wenn wir unsere Sprache sprechen, müssen wir in Sprachen schreiben lernen, die Computer verstehen können, wenn wir ihnen sagen wollen, was sie tun sollen.

WELCHE SPRACHEN WIRST DU LERNEN?

Es gibt mittlerweile Hunderte verschiedene Computersprachen. In diesem Buch konzentrieren wir uns auf die drei Sprachen, die dafür sorgen, dass das World Wide Web funktioniert: HTML, CSS und JavaScript. Diese drei Sprachen lassen sich so vielfältig kombinieren, dass man damit Apps für Smartphones, supercoole interaktive Websites und sogar Spiele entwickeln kann, die du auf deinem Computer spielen kannst!

In den Anfängen des Internets bestanden Webseiten nur aus Text. Die **erste Website** wurde im November 1990 veröffentlicht. Die Buchstaben konnten unterschiedlich groß sein, und ab und zu gab es Links zu anderen Webseiten. Das machte eine Computersprache möglich, die HTML heißt.



HTML steht für Hypertext Markup Language, was hauptsächlich bedeutet, dass man mit HTML Text in Links verwandeln kann, aber HTML kann noch viel mehr, wie du bald feststellen wirst.

In dieser ersten Phase gab es keine große Auswahl an Schriftarten oder schicken Layouts. Es gab auch kaum Bilder, und an Animationen war gar nicht erst zu denken!

Niemand dachte daran, sich zu beschweren. In den Anfängen des Internets war es unglaublich aufregend, von einer Seite auf eine andere zu klicken und immer etwas Neues zu entdecken. Noch aufregender war, dass man ohne große Umstände etwas veröffentlichen konnte, das dann jeder mit Internetzugang lesen konnte!

Es dauerte nicht lange, da wurden Bilder, Farben, Formulare und immer neue Gestaltungsmöglichkeiten eingeführt. CSS wurde erfunden, damit man Webseiten ansprechender gestalten konnte. Wenn man weiß, wie es geht, lassen sich damit **interessante geometrische Formen** gestalten.



CSS ist die Abkürzung für Cascading Style Sheets. Mithilfe von CSS lassen sich dem HTML-Code Stilelemente – etwa Farben, Rahmenlinien und Hintergründe – hinzufügen.

Es dauerte nicht lange, da wollten die Leute Spiele spielen, Animationen erstellen und Websites gestalten, die auf Befehle und Klicks reagieren. Sie wollten, dass die Website-Besucher nicht nur lesen und sehen, sondern auch Einfluss darauf nehmen konnten, mit einem Wort – sie wollten Interaktivität!

Damit Webseiten interaktiv sein konnten, wurde JavaScript entwickelt. Immer wenn du eine Webseite besuchst, auf der sich etwas bewegt oder plötzlich erscheint, sodass sich die Seite verändert, steckt wahrscheinlich JavaScript dahinter.

Starte deinen Browser und schau dir ein paar Websites an, die HTML, CSS und JavaScript besonders gelungen kombinieren:

» **Deutsches Museum virtuell** (digital.deutsches-museum.de/virtuell)

Im virtuellen Deutschen Museum kannst du dir ungestört und ganz aus der Nähe Ausstellungen im Deutschen Museum in München anschauen und Erläuterungen anhören, die mithilfe von HTML, CSS und JavaScript für dich zugänglich gemacht wurden.

» **Nordmeerforscher** (www.nordmeerforscher.de)

Nordmeerforscher ist eine interaktive Website für alle, die am Meeresleben interessiert sind. Begib dich auf Unterwassermissionen und lerne dabei eine Menge über das, was unterhalb der Wasseroberfläche los ist. Und das alles basiert auf JavaScript und HTML!

» **Maus- und Tastentraining** (www.toniklix.de)

Diese Website hilft dir dabei, deine Maus und deine Tastatur besser in den Griff zu bekommen. Und das Beste daran ist, dass diese Website mit den Sprachen erstellt worden ist, die du in diesem Buch lernen wirst!

WERKZEUGE ZUSAMMENSUCHEN

Jedes Projekt erfordert besondere Werkzeuge. Das ist beim Schreiben von Computerprogrammen auch nicht anders. Statt Hammer oder Bleistift verwenden JavaScript-Programmierer Werkzeuge, die sie im Internet finden.

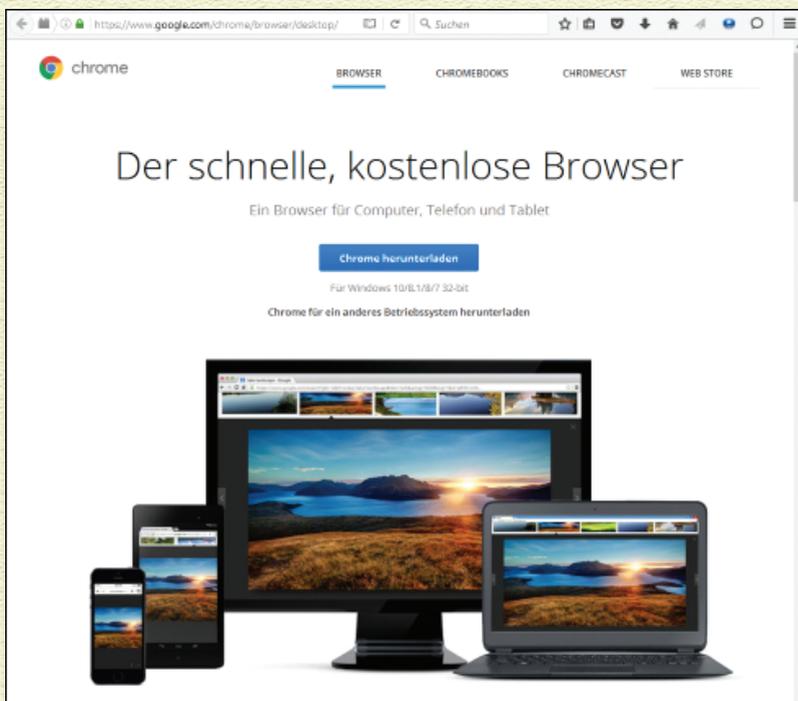
Lass uns jetzt mal deinen Werkzeugkasten zusammenstellen.

DEN BROWSER VORBEREITEN

Das wichtigste Werkzeug für die Arbeit mit JavaScript ist ein Browser. Du hast verschiedene Browser zur Auswahl, und mit fast allen kannst du JavaScript ausführen. Wahrscheinlich hast du bereits einen Browser auf deinem Computer installiert.

Die am häufigsten verwendeten Browser sind heutzutage Mozilla Firefox, Apple Safari, Google Chrome und Microsoft Internet Explorer. Für dieses Buch haben wir uns für Chrome entschieden, weil er momentan der beliebteste Browser ist und schon eine große Zahl nützlicher Werkzeuge für die Arbeit mit JavaScript mitbringt.

Wenn du Chrome noch nicht installiert hast, musst du den **Browser im Internet downloaden** und installieren. Das kannst du mit jedem anderen Browser machen. Gib die Webadresse www.google.com/chrome/browser/desktop in die Adresszeile ein und folge den Installationsanweisungen, die du auf der Seite findest. Wenn die Installation abgeschlossen ist, startest du Chrome.



MIT JSFIDDLE ARBEITEN

Jetzt wird es Zeit, dass du unseren JavaScript-Lieblingsspielplatz kennlernst: JSFiddle. Statt mit Schaukeln, Rutschen und Sandkasten kannst du dort mit JavaScript-Anweisungen, HTML-Tags und CSS-Styles herumspielen.

JSFiddle ermöglicht dir, mit deinem Browser Code zu schreiben und damit zu experimentieren. Du kannst dort Code ausprobieren, herausfinden, was andere zu deinen Programmierungen sagen, Code mit anderen austauschen und sogar mit deinen Freunden zusammen an Programmen arbeiten!

Wir werden in diesem Buch JSFiddle dazu nutzen, mit HTML, CSS und JavaScript zu experimentieren. Die Seite heißt JSFiddle, weil du dort mit Code »herumspielen« kannst (das englische »to fiddle« bedeutet im Deutschen unter anderem »herumspielen« oder »herumfummeln«). Fangen wir also an herumzuspielen!

Öffne deinen Chrome-Browser und gib die Adresse <http://jsfiddle.net> in die Adresszeile ein.

In JSFiddle gibt es drei Bereiche, in die man drei verschiedene Arten Webcode eingeben kann: HTML, CSS und JavaScript. Der vierte Bereich unten rechts mit der Beschriftung **Result** zeigt die Ergebnisse von dem, was du in die anderen Bereiche eingegeben hast. »Result« ist ein englisches Wort und bedeutet »Ergebnis«. In der linken Spalte (**Fiddle Meta** = Übergeordnetes) kannst du **einen Titel und eine Beschreibung für dein Projekt eingeben**. Die obere Werkzeugleiste hat Schaltflächen, mit denen du den Programmcode ausführen (**Run**), speichern (**Save**) und aufräumen (**Tidy**) kannst.



Die Sprache der Programmierer ist Englisch. Auch JSFiddle gibt es nur auf Englisch. Als Programmierer wirst du also nicht darum herumkommen, englische Wörter zu lernen und zu gebrauchen. In diesem Buch werden wir dir die wichtigsten Wörter übersetzen.



EINEN JSFIDDLE-ACCOUNT ANLEGEN

Bei JSFiddle kann jeder einen Account anlegen und seine Programme mit anderen teilen.

Wenn Programmierer ihre Programme auf JSFiddle veröffentlichen, erklären sie sich damit einverstanden, dass andere ihre Arbeit kopieren, bearbeiten oder neu veröffentlichen. Man sollte jedoch immer so höflich sein, den Originalautor zu erwähnen, wenn man anderer Leute Code verwendet.

Du musst keinen JSFiddle-Account anlegen, um mit diesem Buch weiterzuarbeiten, aber mit Account ist die Prüfung und die Veröffentlichung deiner Arbeit einfacher.

Die Anmeldung bei JSFiddle läuft folgendermaßen ab:

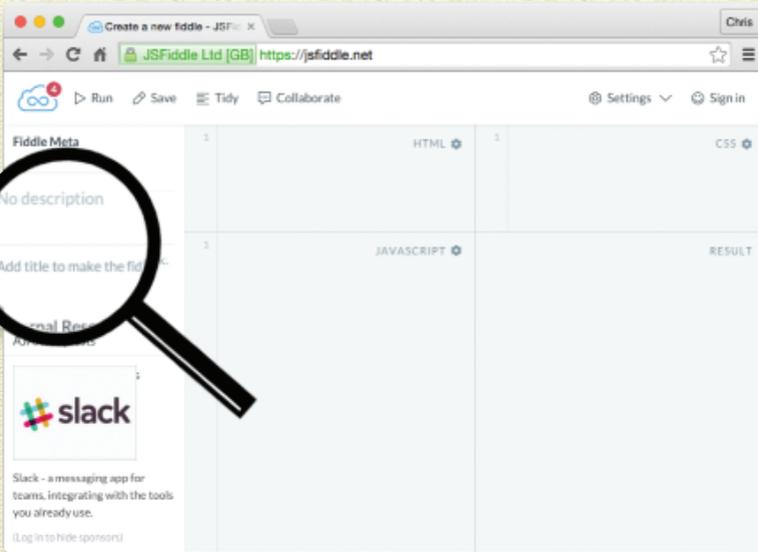
- 1** Klicke auf den Link **Sign in** am oberen rechten Rand des Bildschirms. »Sign in« heißt so viel wie »Anmelden«.
- 2** Klicke unterhalb der Login-Felder auf den blauen Link **Sign up**. »Sign up« bedeutet »Registrieren«.
- 3** Fülle auf der folgenden Seite das Anmeldeformular aus (E-Mail-Adresse, Benutzername und Passwort) und klicke auf **Create an account**.

Nun erscheint die Seite **Edit Your Profile**. Wenn du möchtest, kannst du auf dieser Seite Angaben zu deiner Person machen und speichern. Nötig ist das nicht.

- 4** Wenn du so weit bist, klickst du auf den Link **Editor** oben auf der Seite. Jetzt kann das Spiel beginnen.



Die Bereiche in JSFiddle lassen sich in der Größe verändern. Klicke einfach auf die Ränder und ziehe sie so, wie es dir am liebsten ist.



Fürs Erste konzentrieren wir uns auf den JavaScript-Bereich. Folge der Anleitung und erstelle dein erstes JSFiddle-Programm:

1 Gib im JavaScript-Bereich unten links die folgende JavaScript-Anweisung ein:

```
alert ("Hallo Leute!");
```

2 Klicke auf die Schaltfläche **Run** in der oberen Werkzeugleiste.

Sofort erscheint ein Pop-up-Fenster mit der Botschaft »Hallo Leute!«

3 Klicke auf **Ok**, um das Pop-up-Fenster zu schließen.

Herzlichen Glückwunsch! Du hast gerade dein erstes JavaScript-Programm geschrieben! JSFiddle kann aber nicht

nur JavaScript-Programme laufen lassen. Du kannst hier auch die HTML- und CSS-Bereiche nutzen und sie mit deinem JavaScript-Code zusammenarbeiten lassen. Du wirst die einzelnen Bereiche in den folgenden Abschnitten genauer kennenlernen. Aber zunächst schauen wir uns schnell an, was JSFiddle so alles leisten kann.

Der CSS-Bereich liegt in JSFiddle im oberen rechten Bildschirmbereich. Mit CSS kannst du zum Beispiel das Aussehen von Texten und Grafiken verändern. Wenn du die Schriftfarbe auf deiner Seite ändern willst, machst du das mit CSS.

Die folgende Anleitung zeigt dir, wie du eines unserer Programme verändern kannst:

- 1 Gib die Adresse <http://jsfiddle.net/JavaScriptJunior/u2q2fyfz> in die Adresszeile deines Browsers ein.**



Achte darauf, dass du die Internetadresse ganz genau eintippst und die Großbuchstaben an der richtigen Stelle sind.

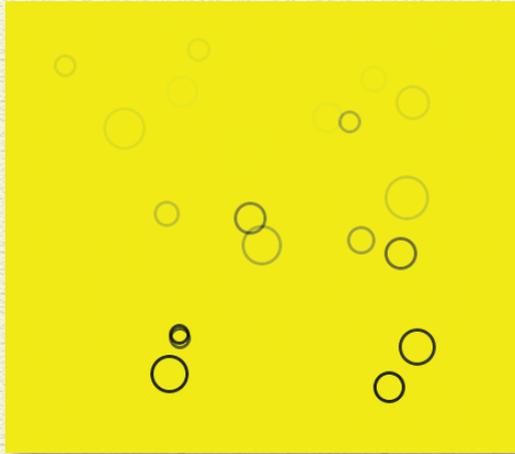
- 2 Schau dir die vier Bereiche auf dem Bildschirm näher an: Drei davon enthalten Code, und im **Result**-Bereich läuft eine **Animation mit aufsteigenden Blasen**.**

Hast du eine Ahnung, was die einzelnen Codeteile bewirken?

- 3 Schau dir den CSS-Bereich genauer an: Du siehst dort drei Codeabschnitte.**



Wenn du nicht den kompletten Code sehen kannst, kannst du auf den Rand des CSS-Bereichs klicken und ihn größer ziehen.



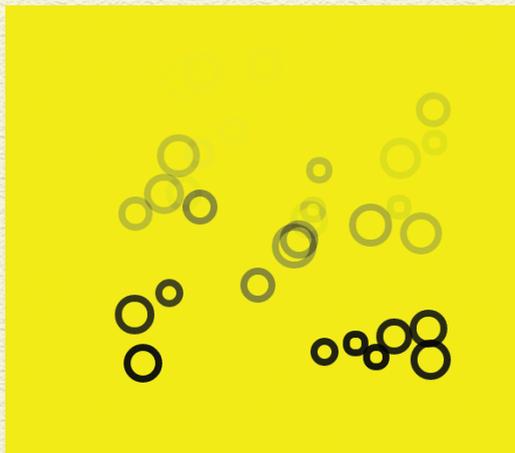
- 4** Suche den Code »`border:3px solid black;`« und ändere ihn zu `border:8px solid black;`.



Achte darauf, dass du deine Eingabe nicht mit einem Punkt beendest. Punkte beenden einen Satz, CSS-Zeilen werden meist mit einem Semikolon beendet.

- 5** Klicke auf die Schaltfläche **Run** in der oberen Werkzeugleiste, um die Animation neu zu starten.

Die Kreise haben jetzt viel **breitere Ränder**.



»Border« ist das englische Wort für »Rand« oder »Begrenzung«. Wenn du darüber nachdenkst, was du gerade gemacht hast und wie es sich ausgewirkt hat, was denkst du, was das Wort »solid« im Code bewirkt? Um das herauszufinden, kannst du Folgendes ausprobieren:

1 Ändere im CSS-Bereich den ersten Wert nach »border:« wieder in eine kleinere Zahl ab (etwa 2 oder 3) und klicke auf **Run**.

Die Kreisränder werden wieder schmaler.

2 Ändere das Wort »solid« nach dem Wort »border:« in eines der folgenden Worte ab (»solid« heißt auf Deutsch »voll« oder »massiv«):

- » dotted (»gepunktet«)
- » dashed (»gestrichelt«)
- » double (»doppelt«)
- » groove (»Rille«)
- » ridge (»Berggrat«)
- » inset (»versenkt«)
- » outset (»erhöht«)

3 Klicke auf **Run** und schau, was passiert.

Das Wort legt den Stil der Kreisränder fest!

Schau dir jetzt einmal den dritten Wert nach »border:« an, der momentan **black** ist. (Das englische Wort »black« kennst du sicherlich schon. Es bedeutet »schwarz«.) Der dritte Wert steuert die Farbe der Kreise.

Die folgende Anleitung zeigt, wie du die Farbe der Kreise verändern kannst:

- 1 Wähle deine Lieblingsfarbe, zum Beispiel Blau (blue), Rot (red) oder Grün (green).**
- 2 Ersetze den Farbnamen »black« im CSS-Bereich durch deinen Farbnamen.**
- 3 Klicke auf Run.**

Die Kreise haben jetzt deine Lieblingsfarbe!



Eine umfassende Liste mit Farbnamen, die dein Computer versteht, findest du unter der Webadresse <https://wiki.selfhtml.org/wiki/Farbnamen>.

Jetzt wenden wir uns dem HTML-Bereich oben links zu. Im Vergleich zum CSS-Bereich und zum JavaScript-Bereich enthält er nicht besonders viel.

Mit HTML gibt man Webseiten eine Struktur und stellt Container zur Verfügung, in denen JavaScript-Programme laufen können. Bei unserem kleinen Blubber-Programm stellt der HTML-Code nur den Platz auf der Seite bereit, auf der die Kreise blubbern können.

Aber mit HTML kann man noch viel mehr machen. Die folgende Anleitung zeigt, wie du unser Blubber-Programm ein wenig verändern kannst:

- 1 Setze deinen Cursor im HTML-Bereich hinter `</div>` und drücke , um in die nächste Zeile zu springen.**
- 2 Gib dort Folgendes ein:**

```
<h1>blubb blubb</h1>
```

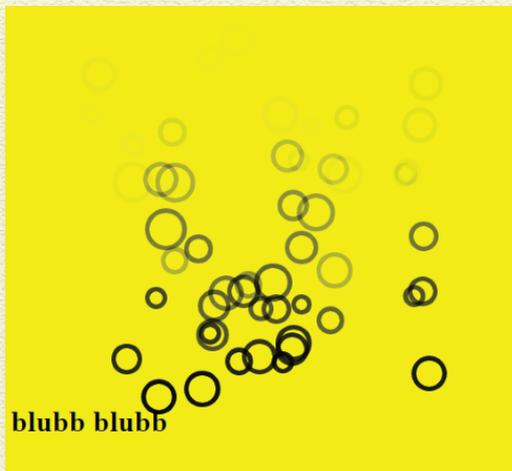
Dein HTML-Bereich sollte nun den folgenden Code enthalten:

```
<div id="o"></div>  
<h1>blubb blubb</h1>
```



HTML-Code umgibt normalen Text mit besonderem Code. Wenn man Text auf diese Weise mit Code ergänzt, nennt man das den Text auszeichnen. HTML ist eine Abkürzung und steht für Hypertext Markup Language, auf Deutsch Hypertext-Auszeichnungssprache.)

- 3** Klicke auf **Run** und schau dir das **Ergebnis** im **Result-Bereich** an.



In HTML werden `<h1>` und `</h1>` *Tags* genannt. Die Tags vor und nach Text sagen dem Browser, dass der eingeschlossene Text etwas Besonderes ist. `<h1>` bezeichnet zum Beispiel eine Überschrift der höchsten Ebene, die größte und wichtigste Überschrift auf einer Seite.

Ein weiterer nützlicher Tag ist `<p>`. Er markiert einen Absatz. Einen `<p>`-Tag fügt man folgendermaßen ein:

- 1 Setze den Cursor hinter den `</h1>`-Tag und drücke , um in die nächste Zeile zu springen.**
- 2 Gib `<p>` ein und tippe dann den Text, der angezeigt werden soll.**

JSFiddle erstellt automatisch den abschließenden `</p>`-Tag.

- 3 Klicke auf **Run** und schau dir den Text im **Result**-Bereich an.**

Die richtig interessanten Sachen spielen sich im JavaScript-Bereich im unteren linken Teil von JSFiddle ab.

- 1 Suche im JavaScript-Bereich die Zeile, in der »max = 36« steht, und ändere sie in `max = 80`.**
- 2 Klicke auf **Run**.**

Viele Kreise, aber nicht alle, sind jetzt größer als vorher.

Wenn du siehst, wie sich die Kreise verhalten, wenn du den Wert von `max` veränderst, kannst du dann ahnen, was passiert, wenn du die Zeile »`min = 12`« veränderst? Probiere es aus, dann wirst du sehen, ob du Recht hattest! Wenn du der Meinung warst, dass `max` die maximale Größe der Kreise und `min` die minimale Größe der Kreise festlegt, dann trifft das den Nagel auf den Kopf.

Die nächsten beiden Zeilen im JavaScript-Bereich sind »`blasen = 100`« und »`zeitVerzögerung = 8000`«.

Spiele mit diesen Werten und schau, was passiert. Du wirst feststellen, dass »`blasen = 100`« festlegt, wie viele Kreise auftauchen sollen, und »`zeitVerzögerung`« etwas damit zu tun hat, wie schnell die Kreise nach oben steigen.

DEINEN FIDDLE SPEICHERN UND VERÖFFENTLICHEN

Nun ist es Zeit, deine Version unseres Blubber-Programms deinen Freunden zugänglich zu machen!

1 **Klicke am linken Rand auf [Fiddle Meta](#).**

Es öffnet sich ein Textfeld: »Kapitel 1 – Blubberblasen«.

2 **Überschreibe den Text mit einem Titel deiner Wahl.**

3 **Klicke in der oberen Menüleiste auf [Fork](#).**

Wie das Symbol auf der Schaltfläche schon andeutet, bedeutet »fork« so viel wie »gabeln«. Dabei wird eine Kopie des ursprünglichen Programms erstellt und deinem Account zugeordnet. Du kannst jetzt damit weiterarbeiten.

Bei eigenen Programmen, die du nicht von anderen Autoren übernommen hast, klickst du stattdessen auf [Save](#) (Speichern).

4 **Klicke nun rechts oben auf deinen Benutzernamen und dann auf [Dashboard](#).**

Hier findest du alle deine Fiddles. Wenn du ein Fiddle deinen Freunden zeigen möchtest, klickst du es an und schickst deinen Freunden die Adresse aus dem Adressfenster.

ZUSAMMENFASSUNG

In diesem Kapitel hast du gelernt, was Programmierer tun, und du hast erste Experimente mit Code in JSFiddle gemacht. Im nächsten Kapitel schauen wir uns die Programmiersprache JavaScript genauer an und erklären dir die besonderen Regeln, die man beachten muss, wenn man JavaScript schreibt.