

HANS-GEORG SCHUMANN



PROGRAMMIEREN LERNEN OHNE VORKENNTNISSE



Hinweis des Verlages zum Urheberrecht und Digitalen Rechtemanagement (DRM)

Liebe Leserinnen und Leser,

dieses E-Book, einschließlich aller seiner Teile, ist urheberrechtlich geschützt. Mit dem Kauf räumen wir Ihnen das Recht ein, die Inhalte im Rahmen des geltenden Urheberrechts zu nutzen. Jede Verwertung außerhalb dieser Grenzen ist ohne unsere Zustimmung unzulässig und strafbar. Das gilt besonders für Vervielfältigungen, Übersetzungen sowie Einspeicherung und Verarbeitung in elektronischen Systemen.

Je nachdem wo Sie Ihr E-Book gekauft haben, kann dieser Shop das E-Book vor Missbrauch durch ein digitales Rechtemanagement schützen. Häufig erfolgt dies in Form eines nicht sichtbaren digitalen Wasserzeichens, das dann individuell pro Nutzer signiert ist. Angaben zu diesem DRM finden Sie auf den Seiten der jeweiligen Anbieter.

Beim Kauf des E-Books in unserem Verlagsshop ist Ihr E-Book DRM-frei.

Viele Grüße und viel Spaß beim Lesen,

Dhr mitp-Verlagsteam





Hans-Georg Schumann

JAVASCRIPT FÜR KIDS

PROGRAMMIEREN LERNEN OHNE VORKENNTNISSE



Bibliografische Information der Deutschen Nationalbibliothek

Die Deutsche Nationalbibliothek verzeichnet diese Publikation in der Deutschen Nationalbibliografie; detaillierte bibliografische Daten sind im Internet über <http://dnb.d-nb.de> abrufbar.

ISBN 978-3-7475-0264-8 1. Auflage 2020

www.mitp.de E-Mail: mitp-verlag@sigloch.de Telefon: +49 7953 / 7189 - 079 Telefax: +49 7953 / 7189 - 082

© 2020 mitp-Verlags GmbH & Co. KG, Frechen

Dieses Werk, einschließlich aller seiner Teile, ist urheberrechtlich geschützt. Jede Verwertung außerhalb der engen Grenzen des Urheberrechtsgesetzes ist ohne Zustimmung des Verlages unzulässig und strafbar. Dies gilt insbesondere für Vervielfältigungen, Übersetzungen, Mikroverfilmungen und die Einspeicherung und Verarbeitung in elektronischen Systemen.

Die Wiedergabe von Gebrauchsnamen, Handelsnamen, Warenbezeichnungen usw. in diesem Werk berechtigt auch ohne besondere Kennzeichnung nicht zu der Annahme, dass solche Namen im Sinne der Warenzeichen- und Markenschutz-Gesetzgebung als frei zu betrachten wären und daher von jedermann benutzt werden dürften.

Lektorat: Katja Völpel Korrektorat: Petra Heubach-Erdmann Covergestaltung: Christian Kalkert, Sandrina Dralle Covergrafik: Adobe Stock/nadiinko Satz: III-satz, Husby, www.drei-satz.de

INHALT

EINLEITUNG	11
Was heisst Programmieren?	12
Was ist eine Entwicklungsumgebung?	12
Warum gerade JavaScript?	13
Die Entwicklungsumgebung	13
Wie arbeite ich mit diesem Buch?	14
Was brauchst du für dieses Buch?	15
Hinweise für Lehrer	16
ALLER ANFANG IST SCHWER? EINSTIEG IN JAVASCRIPT Ein erstes Hallo, schlicht und einfach	17 17
Ein erstes Hallo, schlicht und einfach	17
Visual Studio Code starten	21
Die nötigen Erweiterungen	25
Der steinige Weg zum zweiten Hallo	28
Visual Studio Code beenden	32
Zusammenfassung	33
Ein paar Fragen	34
aber nach keine Aufaabe	~ .

WENN UND DANN UND MATHE: BEDINGUNG UND KONTROLLE	35	9
Willkommen in JavaScript	36	
Die if-Struktur	41	
if und else	46	
Ein bisschen Grundrechnen	49	
Was für Zahlen?	52	
Zusammenfassung	55	
Ein paar Fragen	56	
und ein paar Aufgaben	56	

1

3	ZENSUREN UND ZAHLENRATEN: VERGLEICHEN UND WIEDERHOLENZensurenbildPunkt für PunktEin kleines RatespielDein Computer zählt mitNoch mehr SpielkomfortZusammenfassungEin paar Fragen und ein paar Aufgaben	57 58 62 66 71 73 74 75 75
4	GELD-SPIELEREIEN: NOCH MEHR SCHLEIFEN Spiel mit dem Glück Die for-Struktur Auf dem Weg zum Millionär Macht Lotto wirklich reich? Zeichen-Verkettung Zusammenfassung Ein paar Fragen und ein paar Aufgaben	77 77 80 82 89 93 95 96 96
5	MARKE EIGENBAU: FUNKTIONENJavaScript ist lernfähigLokal oder global?TauschprozesseZahlen sortierenZusammenfassungEin paar Fragen und ein paar Aufgaben	97 97 101 104 109 112 112 112
6	FRANKENSTEINS LABOR: KLASSEN UND MODULE Ein neues Baby? Vererbung Programm-Module Zusammenfassung Ein paar Fragen aber keine Aufgabe	113 114 118 123 128 129 129

MEHR ÜBER HTML: BUTTONS UND LABELS Ein Gerüst in HTMLEs passiert etwasNoch mal HalloNoch etwas FeinschliffZusammenfassungEin paar Fragen und eine Aufgabe	131 132 136 140 144 147 148 148	7
HALLO, WIE GEHT'S? KOMPONENTENSAMMLUNG Kleine Button-Parade Listenwahl Von Pünktchen und Häkchen Zusammenfassung Ein paar Fragen und ein paar Aufgaben	149 149 152 156 160 162 163 163	8
WER WEISS WAS? QUIZ-PROJEKT TEIL 1 Zuerst der Plan und dann der Bau Spielbereit? Datentransfer Auf ein Neues Zusammenfassung Ein paar Fragen aber keine Aufgabe	165 165 170 172 176 180 180 180	9
SPIELEN UND LERNEN: QUIZ-PROJEKT TEIL 2 Richtig oder falsch? Aufgabenkontrolle Antwort als Optionen Vokabeln lernen? Zusammenfassung Ein paar Fragen und ein paar Aufgaben	181 186 190 192 196 197 197	10

		min
	JETZT WIRD ES BUNT: GRAFIK IN JAVASCRIPT	199
	Von Punkten und Koordinaten	200
	Das erste Bild	202
	Jetzt wird's bunt	205
	Eckig und rund	208
	Mit Text geht auch	209
	Farbtupfer	211
	Selber zeichnen?	212
	Zusammenfassung	216
	Ein paar Fragen	217
	und ein paar Aufgaben	217
	ΡΠΛΕΡ ΙΕΡΝΕΝ ΙΛΠΕΕΝ, ΑΝΙΜΑΤΙΑΝΕΝ	010
12		219
	Erst mal (irgend)ein Bild	219
	Kommen und genen	221
		226
		232
	Zusammeniassung	234
	Lill paal Flagen	200
		235
17	BUNTES TRIO: KLEINE SPIELESAMMLUNG	237
15	Wie viele Augen?	237
	Schere – Stein – Papier	240
	Auf ein Wort	244
	Angst vor dem Galgen?	247
	Komplettierung	251
	Zusammenfassung	254
	Keine Fragen	255
	aber ein paar Aufgaben	255
	VON VER AUGEL ZUM PLATER; EIN ALEINES BALLSPIEL IEIL I	257
	Ein Ball auf dem Spielfeld	258
	Grenzkontrollen	261
	Player-Klasse	264
	/wei Paddles	267

Der Computer spielt mit

Zusammenfassung

Ein paar Fragen ...

... und eine Aufgabe

PING UND PONG: EIN KLEINES BALLSPIEL TEIL 2 Spiel-Intelligenz? Manuelle Paddle-Steuerung Feintuning Das Gesamt-Listing Zusammenfassung Keine Fragen und keine Aufgaben	275 275 280 283 286 290 290 290	15
SCHLANGE UND KÄFER: DAS SNAKE-SPIEL Ein Feld, zwei Spieler Die Schlange bewegt sich Länger und länger Verfeinerungen Das gesamte Listing Zusammenfassung Eine Frage und eine Aufgabe	291 296 299 303 305 308 308 308	16
TRAU DICH WAS: DODGE ODER HIT? Alter Quelltext, neue Idee? Stand, Duck, Jump Ausweichmanöver Zusammenfassung Keine Fragen und nur eine Aufgabe	309 309 313 316 322 322 322	17
ANHANG A VISUAL STUDIO CODE INSTALLIEREN	323	A
ANHANG B KLEINE CHECKLISTE	329	B
STICHWORTVERZEICHNIS	331	

Für Janne, Julia, Daniel und Katrin



Computer sind schon wahre Wunder-Maschinen. Doch das sind sie nur dadurch, dass es schlaue Programmierer gibt, die die passenden Anwendungen und Spiele erstellen. Wenn du eines Tages zur Gilde der Programmierer zählen willst, dann hast du hier die Möglichkeit, bei null anzufangen – ohne irgendwelche Programmier-kenntnisse.

Hier lernst du, in JavaScript zu programmieren, von dem du sicher schon einmal etwas gehört hast. Das ist nicht die kleine Schwester der Programmiersprache Java. Was ist der Unterschied? Grob gesagt: Es gibt eine Gruppe von Sprachen, da wird aus dem, was du programmierst, direkt ein ausführbares Programm erzeugt. Dazu gehören Java, C#, C++. Und es gibt Sprachen, da ist immer ein sogenannter Interpreter nötig, der das Programm ausführt. Eine solche Sprache ist JavaScript, und der passende Interpreter steckt in fast jedem Webbrowser.

Du hast bestimmt auch schon mal von HTML gehört. Ohne diese »HyperText Markup Language« könntest du dir keine Webseiten anschauen. HTML dient dazu, elektronische Dokumente zu erstellen. Am bekanntesten wurde HTML als das Format für Web-Dokumente, wie z.B. Seiten von Homepages, die sich in einem passenden Browser darstellen lassen.

Und JavaScript, das Mitte der 1990er Jahre entwickelt wurde, erweitert die Möglichkeiten von HTML stark. Mithilfe von JavaScript lassen sich sogar Spiele erstellen, die man direkt im Browser spielen kann. JavaScript hat viel von anderen (großen) Programmiersprachen übernommen. Diese Sprache ist leicht erlernbar, und weil JavaScript-Programme direkt in fast jedem Browser funktionieren, braucht man nicht viel Werkzeug, um Anwendungen und Spiele zu erstellen.

WAS HEISST PROGRAMMIEREN?

Wenn du aufschreibst, was ein Computer tun soll, nennt man das **Programmieren**. Das Tolle daran ist, dass du selbst bestimmen kannst, was getan werden soll. Lässt du dein Programm laufen, macht der Computer das, was du ausgeheckt hast. Natürlich wird er nicht dein Zimmer aufräumen und dir auch keine Tasse Kakao ans Bett bringen. Aber beherrschst du erst mal das Programmieren, kannst du den Computer sozusagen nach deiner Pfeife tanzen lassen.

Allerdings passiert es gerade beim Programmieren, dass der Computer nicht so will, wie du es gerne hättest. Meistens ist das ein Fehler im Programm. Das Problem kann aber auch irgendwo anders im Computer oder im Betriebssystem liegen. Das Dumme bei Fehlern ist, dass sie sich gern so gut verstecken, dass die Suche danach schon manchen Programmierer zur Verzweiflung gebracht hat.

Vielleicht hast du nun trotzdem Lust bekommen, das Programmieren zu erlernen. Dann brauchst du ja nur noch eine passende Entwicklungsumgebung, und schon kann es losgehen.

WAS IST EINE ENTWICKLUNGSUMGEBUNG?

Um ein Programm zu erstellen, musst du erst einmal etwas eintippen. Das ist wie bei einem Brief oder einer Geschichte, die man schreibt. Das Textprogramm dafür kann sehr einfach sein, weil es ja nicht auf eine besondere Schrift oder Darstellung ankommt wie bei einem Brief oder einem Referat. So etwas wird **Editor** genannt.

Ist das Programm eingetippt, kann es der Computer nicht einfach lesen und ausführen. Jetzt muss es so übersetzt werden, dass der PC versteht, was du von ihm willst. Weil er aber eine ganz andere Sprache spricht als du, muss ein Dolmetscher her.

Ein solcher Dolmetscher (= Interpreter) steckt in jedem Browser, JavaScript funktioniert sogar unter mehreren Betriebssystemen. Dein Computer kann also ein Windows-PC oder ein Linux-PC sein, ein Android-Smartphone oder irgendein anderes System. Ein und dasselbe JavaScript-Programm kann so (eventuell mit kleinen Abweichungen) auf fast jedem Computer oder Smartphone funktionieren. Schließlich müssen Programme getestet, überarbeitet, verbessert, wieder getestet und weiterentwickelt werden. Da ist man froh, wenn man einige zusätzliche Hilfen hat. Daraus wird dann ein ganzes System, die Entwicklungsumgebung.

WARUM GERADE JAVASCRIPT?

Leider kannst du nicht so programmieren, wie dir der Schnabel gewachsen ist. Eine **Programmiersprache** muss so aufgebaut sein, dass möglichst viele Menschen in möglichst vielen Ländern einheitlich damit umgehen können.

Weil in der ganzen Welt Leute zu finden sind, die wenigstens ein paar Brocken Englisch können, besteht auch fast jede Programmiersprache aus englischen Wörtern. Es gab auch immer mal Versuche, z.B. in Deutsch zu programmieren, aber meistens klingen die Wörter dort so künstlich, dass man lieber wieder aufs Englische zurückgreift.

Eigentlich ist es egal, welche Programmiersprache du benutzt. Am besten eine, die möglichst leicht zu erlernen ist. Wie du weißt, bekommst du es in diesem Buch mit der Programmiersprache JavaScript zu tun, die mittlerweile sehr weit verbreitet ist. (Willst du mal in andere Sprachen hineinschnuppern, dann empfehle ich dir z.B. eines der anderen Kids-Bücher über Python, Java oder C++.)

Der Weg zum guten Programmierer kann ganz schön steinig sein. Nicht selten kommt es vor, dass man die Lust verliert, weil einfach gar nichts klappen will. Das Programm tut etwas ganz anderes, man kann den Fehler nicht finden und man fragt sich: Wozu soll ich eigentlich programmieren lernen, wo es doch schon genug Programme gibt?

Gute Programmierer werden immer gesucht, und dieser Bedarf wird weiter steigen. Und JavaScript gehört dabei durchaus zu den erwünschten Sprachen. Wirklich gute Programmierer werden auch wirklich gut bezahlt. Es ist also nicht nur einen Versuch wert, es kann sich durchaus lohnen, das Programmieren in JavaScript zu erlernen.

DIE ENTWICKLUNGSUMGEBUNG

Um eine Entwicklungsumgebung für JavaScript musst du dich nicht weiter kümmern, wenn dir eine einfache reicht. Das ist der Editor, der mit einem Betriebssystem wie Windows mitinstalliert wird. Besser aber ist etwas mit mehr Komfort, wie Visual Studio Code von Microsoft. Dieses Paket werden wir hier ausgiebig benutzen. Du kannst es dir von dieser Seite herunterladen:

https://code.visualstudio.com/

(Im Anhang steht, wie du dieses Paket installierst, und im Buch wird erklärt, wie du es nutzen kannst.)

UND WAS BIETET DIESES BUCH?

Über eine ganze Reihe von Kapiteln verteilt lernst du

- das Basiswissen von JavaScript kennen
- einiges über HTML und DOM
- etwas über objektorientierte Programmierung
- die grafischen Möglichkeiten von JavaScript kennen
- ◎ wie man eigene Game- und Player-Module programmiert

Im Anhang gibt es dann noch zusätzliche Informationen und Hilfen, unter anderem über Installationen und den Umgang mit Fehlern.

WIE ARBEITE ICH MIT DIESEM BUCH?

Grundsätzlich besteht dieses Buch aus einer Menge Text mit vielen Abbildungen dazwischen. Natürlich habe ich mich bemüht, alles so zuzubereiten, dass daraus lauter gut verdauliche Happen werden. Damit das Ganze noch genießbarer wird, gibt es zusätzlich noch einige Symbole, die ich dir hier gern erklären möchte:

ARBEITSSCHRITTE

Wenn du dieses Zeichen siehst, heißt das: Es gibt etwas zu tun. Damit kommen wir beim Programmieren Schritt für Schritt einem neuen Ziel immer näher.

Grundsätzlich lernt man besser, wenn man einen Programmtext selbst eintippt oder ändert. Aber nicht immer hat man große Lust dazu. Deshalb gibt es alle Projekte im Buch auch als Download:

http://www.mitp.de/0263

Und hinter einem Programmierschritt findest du auch den jeweiligen Namen des Projekts oder eines Ordners (z.B. \rightarrow PROJEKT1). Wenn du also das Projekt nicht selbst erstellen willst, kannst du stattdessen die passenden Dateien laden (sie liegen alle im Haupt-Ordner PROJEKTE).

AUFGABEN

Am Ende eines Kapitels findest du jeweils eine Reihe von Fragen und Aufgaben. Diese Übungen sind nicht immer ganz einfach, aber sie helfen dir, noch besser zu programmieren. Lösungen zu den Aufgaben findest du in verschiedenen Formaten ebenfalls im Verzeichnis PROJEKTE. Du kannst sie dir alle im Browser oder in einem Textverarbeitungsprogramm anschauen. Oder du lässt sie dir ausdrucken und hast sie dann schwarz auf weiß, um sie neben deinen Computer zu legen. (Auch die Programme zu den Aufgaben liegen im Ordner PROJEKTE.)

NOTFÄLLE

Vielleicht hast du irgendetwas falsch gemacht oder etwas vergessen. Oder es wird gerade knifflig. Dann fragst du dich, was du nun tun sollst. Bei diesem Symbol findest du eine Lösungsmöglichkeit. Notfalls kannst du aber auch ganz hinten im Anhang B nachschauen, wo einige Hinweise zur Pannenhilfe aufgeführt sind.

WICHTIGE STELLEN IM BUCH

Hin und wieder findest du ein solch dickes Ausrufezeichen im Buch. Dann ist das eine Stelle, an der etwas besonders Wichtiges steht.

Wenn du ein solches »Wow« siehst, geht es um ausführlichere Informationen zu einem Thema.

WAS BRAUCHST DU FÜR DIESES BUCH?

Installiert wird Visual Studio Code mit einem Setup-Programm in ein Verzeichnis deiner Wahl. Außerdem solltest du einen Ordner einrichten, in dem du später deine JavaScript-Projekte unterbringen kannst – z.B. D:\JSCRIPT.

Die Beispielprogramme in diesem Buch gibt es alle als Download von der Homepage des Verlages, falls du mal keine Lust zum Abtippen hast:

http://www.mitp.de/0263

Und auch die Lösungen zu den Fragen und Aufgaben sind dort untergebracht (alles im Ordner PROJEKTE).

BETRIEBSSYSTEM

Die meisten Computer arbeiten heute mit dem Betriebssystem Windows. Davon brauchst du eine möglichst neue Version (am besten von Windows 10). JavaScript gibt es unter anderem auch für Linux, aber hier im Buch geht es nur um Windows.





SPEICHERMEDIEN

Auf jeden Fall benötigst du etwas wie einen USB-Stick oder eine SD-Card, auch wenn du deine Programme auf die Festplatte speichern willst. Auf einem externen Speicher sind deine Arbeiten auf jeden Fall zusätzlich sicher aufgehoben. Bitte gegebenenfalls deine Eltern oder Lehrer um Hilfe.

HINWEISE FÜR LEHRER

Dieses Buch versteht sich auch als Lernwerk für den Informatik-Unterricht in der Schule. Dort setzt natürlich jeder Lehrer seine eigenen Schwerpunkte. Benutzen Sie an Ihrer Schule bereits ein Werk aus einem Schulbuchverlag, so lässt sich dieses Buch auch als Materialienband einsetzen – in Ergänzung zu dem vorhandenen Schulbuch. Weil dieses Buch sozusagen »bei null« anfängt, ist ein direkter Einstieg in JavaScript möglich – ohne irgendwelche anderen Programmierkenntnisse.

Ein wichtiger Schwerpunkt in diesem Buch ist die objektorientierte Programmierung (OOP). Auf die wichtigsten Eigenheiten (Kapselung, Vererbung und Polymorphie) wird ausführlich eingegangen. Ein großer Schwerpunkt ist die Programmierung von Spielen.

In den Projekten werden alle wesentlichen Elemente des Wortschatzes von HTML und JavaScript wie auch die wichtigsten Grafik-Komponenten eingesetzt. In den Lösungen zu den Aufgaben finden Sie weitere Vorschläge zur Programmierung.

ÜBUNGSMEDIEN

Für den Informatik-Unterricht sollte jeder Schüler ein eigenes externes Speichermedium haben, um darauf seine Programmierversuche zu sichern. So wird verhindert, dass sich auf der Festplatte des Schulcomputers mit der Zeit allerlei »Datenmüll« ansammelt. Außerdem dient der eigene Datenträger dem Datenschutz: Nur der betreffende Schüler kann seine Daten manipulieren.

AUF DIE DATEIEN ZUM BUCH VERZICHTEN?

Vielleicht ist es Ihnen lieber, wenn Ihre Schüler die Projekte alle selbst erstellen. Dann lassen Sie die Download-Dateien einfach (erst einmal) weg.

REGELMÄBIG SICHERN

Es kann nicht schaden, die Programmdateien, an denen gerade gearbeitet wird, etwa alle zehn Minuten zu speichern. Denn Computer pflegen gern gerade dann »abzustürzen«, wenn man seine Arbeit längere Zeit nicht gespeichert hat.



Du willst gleich loslegen? Dem Computer endlich mal etwas sagen, was er für dich tun kann? Na, dann schalte deinen PC an und lass erst mal Windows auftauchen. Von da aus geht es dann direkt zum ersten Programmprojekt in JavaScript.

In diesem Kapitel lernst du

- etwas über HTML
- wie man ein kleines Programm mit dem Editor schreibt
- ◎ eine Anweisung für die Ausgabe kennen
- ø wie der Browser das Programm ausführen kann
- wie man ein neues Projekt in Visual Studio Code erstellt
- etwas über die Datei LAUNCH.JSON
- wie man Visual Studio Code beendet

EIN ERSTES HALLO, SCHLICHT UND EINFACH

Eigentlich ist es ganz einfach, ein erstes Programm in JavaScript zu schreiben. Du brauchst dazu nur einen simplen Editor wie den, der stets mit Windows mitinstal-

liert wird, auch Notepad genannt. Und du brauchst einen Browser, das ist das Programm, mit dem du im Internet surfst oder Dateien von dort herunterlädst. Also z.B. Chrome von Google oder Edge von Microsoft, um nur zwei zu nennen.

Probieren wir aus, wie wir den Browser zum Hallo-Sagen bringen.

Klicke mit der rechten Maustaste auf den Desktop, ein Kontextmenü öffnet sich, klicke dort auf NEU und dann auf TEXTDOKUMENT.



Auf dem Desktop findest du nun ein neues Symbol.



Wenn du auf dieses Symbol doppelklickst, öffnet sich kurze Zeit später der zugehörige Editor und bietet dir ein leeres Fenster an.



Bevor wir da etwas hineintippen, speichern wir die noch leere Datei unter einem passenden Namen:

≫ Klicke im Editor oben auf DATEI und dann auf SPEICHERN UNTER.

📗 Neues Textdokumer	nt.txt - Editor			⊐ ×
Datei Bearbeiten For	mat Ansicht Hilfe			
Neu Neues Fenster Öffnen Speichern Speichern unter	Strg+Umschalt+P Strg+Umschalt+P Strg+C Strg+ Strg+Umschalt+	N N D S S		^
Seite einrichten Drucken Beenden	Strg+	p		
				~
Ze 1, Sp 1	100% Win	dows (CRLF)	UTF-8	.:1

≫ Gib der Datei den Namen INDEX.HTML und klicke dann auf SPEICHERN.

💫 Speichern unter			×
\leftarrow \rightarrow \checkmark \uparrow \blacksquare > Arbeitsplatz	> Desktop > 〜 心	"Desktop" durch	nsuchen 🔎
Datei <u>n</u> ame: index.html			~
Dateityp: Textdateien (*.txt)			~
✓ <u>O</u> rdner durchsuchen	[<u>S</u> peichern	Abbrechen

Warum wir die Datei so genannt haben, erkläre ich später. Jetzt wollen wir sie mit Inhalt füllen. Aber nur mit einigen wenigen Zeilen.

 \gg Tippe also ein:

<script> document.write("Hallo"); </script>

≫ Dann speichere das Ganze – z.B. über DATEI und SPEICHERN oder mit [Strg]+[S].

Und nun kommen wir zu dem, was du als Symbol auf dem Desktop sehen kannst:



Warum INDEX und warum HTML? Die Kennung HTML ist die Abkürzung für »HyperText Markup Language«, eine Hilfssprache für das Programmieren von Webseiten.

Was konkret heißt, dass unser kleines Programm da oben zum Teil in HTML geschrieben wurde. Denn eigentlich ist HTML eine Beschreibungssprache, die den Aufbau einer Webseite beschreibt.

INDEX heißt unsere Datei, weil sie so vom Browser als Startdatei einer vermeintlichen Homepage bzw. Webseite interpretiert wird. Das siehst du gleich nach dem Start.

Wie das Dateisymbol für INDEX.HTML aussieht, hängt davon ab, welcher Browser dein Standard-Browser ist. (Ich benutze außer Microsoft Edge auch Google Chrome und Mozilla Firefox.)

Doppelklicke auf das Symbol mit dem Namen INDEX.HTML.



Der Browser (mit dem du normalerweise im Internet surfst) öffnet sich und zeigt ein einfaches, aber nettes »Hallo«. Womit du dein erstes Programm geschafft hast – allerdings nicht in reinem JavaScript. Das ist ja die Sprache, um die es in diesem Buch eigentlich gehen soll. Lediglich die Zeile in der Mitte ist in JavaScript geschrieben, das Drumherum ist aber nötig, damit der Browser weiß, dass es um JavaScript geht.

Jeder Internet-Browser versteht die Sprache JavaScript, weil er einen eingebauten Interpreter für diese Sprache hat. Ein **Interpreter** ist eine Art Dolmetscher. Allerdings »spricht« ein Browser erst mal nur HTML, eine andere Sprache, in der auch anders programmiert wird als in JavaScript.

Mit der einleitenden Zeile <script> wird dem Browser mitgeteilt, dass gleich etwas kommt, das in einer Scriptsprache, wie man JavaScript auch nennt, verfasst ist. Das abschließende </script> – mit vorgesetztem Schrägstrich – zeigt dem Browser, dass hier die JavaScript-Anweisungen enden.

Die einzige Zeile in JavaScript

document.write("Hallo");

bedeutet, dass in das Browserfenster etwas geschrieben werden soll.

Der entsprechende Text wird in Anführungszeichen gesetzt. Das können sowohl einfache wie auch doppelte sein: 'Hallo' ist also ebenso zulässig wie "Hallo".



Du kannst gerne mal das »Hallo« durch einen anderen (längeren) Text ersetzen, um zu sehen, wie das im Browser aussieht.

VISUAL STUDIO CODE STARTEN

Während der einfache Texteditor am Anfang ausreichen mag, wird er zunehmend unbequem, wenn wir größere Projekte gestalten wollen. Deshalb steigen wir an dieser Stelle um auf einen sogenannten Quelltext-Editor. Ich habe mich für Visual Studio Code von Microsoft entschieden (mit dem man übrigens außer in JavaScript noch in vielen anderen Sprachen programmieren kann).

Bevor wir aber damit arbeiten können, muss Visual Studio Code erst installiert werden. Genaues erfährst du im Anhang A. Hier musst du dir von jemandem helfen lassen, wenn du dir das Einrichten nicht allein zutraust.

Im Folgenden machen wir es uns einfacher und sprechen kurz von VS Code.

Das Allererste aber, was erledigt werden muss, ist das Erzeugen eines Ordners, in dem alle unsere JavaScript-Projekte gespeichert werden sollen. Ich schlage vor, ihn einfach PROJEKTE zu nennen.



Nun können wir VS Code starten. Eine Möglichkeit ist diese:

Doppelklicke auf das Desktop-Symbol mit dem Namen VS CODE. Wenn es nicht vorhanden ist, öffne den Ordner, in dem du VS Code untergebracht hast, und doppelklicke mit der Maus auf das Symbol mit dem Namen CODE.EXE.



Ich empfehle dir, eine Verknüpfung auf dem Desktop anzulegen:

Dazu klickst du mit der rechten Maustaste auf das Symbol f
ür VS Code. Im Kontextmen
ü w
ählst du KOPIEREN.



- Dann klicke auf eine freie Stelle auf dem Desktop, ebenfalls mit der rechten Maustaste. Im Kontextmenü wählst du VERKNÜPFUNG EINFÜGEN.
- Es ist sinnvoll, für das neue Symbol auf dem Desktop den Text CODE.EXE VERKNÜPFUNG durch VS CODE zu ersetzen.

Von nun an kannst du auf das neue Symbol **doppelklicken** und damit VS Code starten.

Je nach Computer kann es eine Weile dauern, bis VS Code geladen ist. Einige Zeit später landest du in einem Willkommen-Fenster.



Ganz oben kann man die Menüleiste von Seite 22 erkennen. Von den Menüs wirst du wahrscheinlich diese vier am meisten benutzen:

- Über das FILE-Menü kannst du Dateien speichern, laden (öffnen), ausdrucken, neu erstellen oder VS Code beenden.
- Die Menüs EDIT und SELECTION helfen dir bei der Bearbeitung deines Programmtextes, aber auch bei anderen Programmelementen. Außerdem kannst du dort bestimmte Arbeitsschritte rückgängig machen oder wiederherstellen.
- Über das RUN-Menü sorgst du dafür, dass dein Projekt ausgeführt wird.
- Und das HELP-Menü bietet dir vielfältige Hilfsinformationen (vor allem auf Englisch) an.

Einige wichtige Menüeinträge sind in einem sogenannten **Popup**-Menü zusammengefasst. Das heißt so, weil es dort aufklappt, wo du gerade mit der **rechten** Maustaste hin klickst.

Und nun schauen wir uns mal unter der Menüleiste um. Lass dich durch den Inhalt dieses Fensters nicht verwirren:



Uns interessiert jetzt das, was (links oben) unter START steht. Bevor wir eine neue Datei erstellen, will VS Code wissen, welchen Ordner wir für unser Programmprojekt ausgewählt haben.

➢ Klicke also auf OPEN FOLDER.



Im Dialogfenster suchst du nun den Unterordner PROJEKTE (den du vorher erstellt hast).

🗙 Open Folder							×
← → × ↑ 🖡 « W	Vork > JScript	t> ~	Ŭ	ر" کر	Script" du	rchsuchen	
Organisieren 🔹 Neuer	Ordner					₩ 50 6,12	?
 Bilder Desktop Dokumente Downloads Musik Videos 	Microsoft VS Code	Projekte					
Ordner:	Projekte		Ore	dner auswä	hlen	Abbrecher	
				aner adowa		, as she chief	·

≫ Klicke abschließend auf ORDNER AUSWÄHLEN.

Das Aussehen des Fensters von VS Code hat sich nun ein wenig geändert:



DIE NÖTIGEN ERWEITERUNGEN

Bevor wir jetzt eine neue Datei erzeugen, benötigen wir das passende Hilfsmittel, damit wir später von VS Code aus direkt den Browser starten können, in dem wir das JavaScript-Programm ausführen wollen.

Klicke dazu oben in der Menüleiste auf RUN und im sich öffnenden Menü ganz unten auf INSTALL ADDITIONAL DEBUGGERS.



Es gibt da ein reichhaltiges Angebot. Für uns interessant sind nur die Erweiterungen für einen der Browser, die wir benutzen, das sind meistens Microsoft Edge, Google Chrome und Mozilla Firefox.



Blättere dich durch das Angebot und suche den Browser aus, den du verwenden willst. Dann klicke dort auf INSTALL.

Kurze Zeit später ist VS Code bereit, deine späteren Programme im entsprechenden Browser zu starten.

Und nun fangen wir endlich an mit unserem Projekt.

➢ Klicke unter Start auf NEW FILE.



Die neue Datei sollte nun gleich unter dem passenden Namen gespeichert werden, damit VS Code weiß, mit welcher Art von Sprache wir arbeiten wollen.

Klicke im Menü auf FILE und dann auf SAVE AS.



Sib den Namen index.html ein und klicke dann auf SPEICHERN.

Save As	×
ightarrow ightarro	'n
Dateiname: index.html Dateityp: HTML (*.html;*.htm;*.shtml;*.xhtml;*.mdoc;*.jsp;*.asp;*.asp;*.aspx;*.jshtm;*.volt)	~
✓ <u>O</u> rdner durchsuchen <u>Speichern</u> Abbreche	n .:

Alternativ kannst du auch ganz links (im EXPLORER) neben PROJEKTE auf das erste Symbol klicken.



]

Hier lässt sich dann auch direkt ein neuer Name eingeben. Den Typ erkennt VS Code an der Endung.



DER STEINIGE WEG ZUM ZWEITEN HALLO

Und nun können wir unseren Programmtext eingeben (auch Quelltext genannt).

Tippe am besten die gleichen Zeilen ein, die wir ganz zu Anfang verwendet haben.

```
<script>
    document.write("Hallo");
</script>
```