

HANS-GEORG SCHUMANN

JAVASCRIPT

FÜR **KIDS**

PROGRAMMIEREN LERNEN
OHNE VORKENNTNISSE



mitp

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,

Ihr mitp-Verlagsteam



Neuerscheinungen, Praxistipps, Gratiskapitel,
Einblicke in den Verlagsalltag –
gibt es alles bei uns auf Instagram und Facebook



[instagram.com/mitp_verlag](https://www.instagram.com/mitp_verlag)



[facebook.com/mitp.verlag](https://www.facebook.com/mitp.verlag)

Inhaltsverzeichnis

Impressum

Einleitung

- Was heißt Programmieren?
- Was ist eine Entwicklungsumgebung?
- Warum gerade JavaScript?
- Die Entwicklungsumgebung
- Wie arbeite ich mit diesem Buch?
- Was brauchst du für dieses Buch?
- Hinweise für Lehrer

Kapitel 1: Aller Anfang ist schwer? Einstieg in JavaScript

- Ein erstes Hallo, schlicht und einfach
- Visual Studio Code starten
- Die nötigen Erweiterungen
- Der steinige Weg zum zweiten Hallo
- Visual Studio Code beenden
- Zusammenfassung
- Ein paar Fragen ...
- ... aber noch keine Aufgabe

Kapitel 2: Wenn und Dann und Mathe: Bedingung und Kontrolle

- Willkommen in JavaScript

Die if-Struktur
if und else
Ein bisschen Grundrechnen
Was für Zahlen?
Zusammenfassung
Ein paar Fragen ...
... und ein paar Aufgaben

Kapitel 3: Zensuren und Zahlenraten: Vergleichen und wiederholen

Zensurenbild
Punkt für Punkt
Ein kleines Ratespiel
Dein Computer zählt mit
Noch mehr Spielkomfort
Zusammenfassung
Ein paar Fragen ...
... und ein paar Aufgaben

Kapitel 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

Kapitel 5: Marke Eigenbau: Funktionen

JavaScript ist lernfähig

Lokal oder global?

Tauschprozesse

Zahlen sortieren

Zusammenfassung

Ein paar Fragen ...

... und ein paar Aufgaben

Kapitel 6: Frankensteins Labor: Klassen und Module

Ein neues Baby?

Vererbung

Programm-Module

Zusammenfassung

Ein paar Fragen ...

... aber keine Aufgabe

Kapitel 7: Mehr über HTML: Buttons und Labels

Ein Gerüst in HTML

Es passiert etwas

Noch mal Hallo

Noch etwas Feinschliff

Zusammenfassung

Ein paar Fragen ...

... und eine Aufgabe

Kapitel 8: Hallo, wie geht's? Komponentensammlung

Kleine Button-Parade

Listenwahl

Von Pünktchen ...

... und Häkchen

Zusammenfassung

Ein paar Fragen ...

... und ein paar Aufgaben

Kapitel 9: 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

Kapitel 10: Spielen und Lernen: Quiz-Projekt Teil 2

Richtig oder falsch?

Aufgabenkontrolle

Antwort als Optionen

Vokabeln lernen?

Zusammenfassung

Ein paar Fragen ...

... und ein paar Aufgaben

Kapitel 11: Jetzt wird es bunt: Grafik in JavaScript

Von Punkten und Koordinaten

Das erste Bild

Jetzt wird's bunt

Eckig und rund

Mit Text geht auch

Farbtupfer

Selber zeichnen?

Zusammenfassung

Ein paar Fragen ...

... und ein paar Aufgaben

Kapitel 12: Bilder lernen laufen: Animationen

Erst mal (irgend)ein Bild

Kommen und gehen

Bildersammlung

Drehungen

Zusammenfassung

Ein paar Fragen ...

... und ein paar Aufgaben

Kapitel 13: Buntes Trio: Kleine Spielesammlung

Wie viele Augen?

Schere - Stein - Papier

Auf ein Wort

Angst vor dem Galgen?

Komplettierung
Zusammenfassung
Keine Fragen ...
... aber ein paar Aufgaben

Kapitel 14: Von der Kugel zum Player: Ein kleines Ballspiel Teil 1

Ein Ball auf dem Spielfeld
Grenzkontrollen
Player-Klasse
Zwei Paddles
Der Computer spielt mit
Zusammenfassung
Ein paar Fragen ...
... und eine Aufgabe

Kapitel 15: Ping und Pong: Ein kleines Ballspiel Teil 2

Spiel-Intelligenz?
Manuelle Paddle-Steuerung
Feintuning
Das Gesamt-Listing
Zusammenfassung
Keine Fragen ...
... und keine Aufgaben

Kapitel 16: 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

Kapitel 17: Trau dich was: Dodge oder Hit?

Alter Quelltext, neue Idee?
Stand, Duck, Jump
Ausweichmanöver
Zusammenfassung
Keine Fragen ...
... und nur eine Aufgabe

Anhang A: Visual Studio Code installieren

Anhang B: Kleine Checkliste

Hans-Georg Schumann

JavaScript für Kids

**Programmieren lernen ohne
Vorkenntnisse**



Impressum

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 978-3-7475-0265-5

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

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
electronic **publication**: Ill-satz, Husby, www.drei-satz.de

Dieses Ebook verwendet das ePub-Format und ist optimiert für die Nutzung mit dem iBooks-reader auf dem iPad von Apple. Bei der Verwendung anderer Reader kann es zu Darstellungsproblemen kommen.

Der Verlag räumt Ihnen mit dem Kauf des ebooks das Recht ein, die Inhalte im Rahmen des geltenden Urheberrechts zu nutzen. 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 Einspeicherung und Verarbeitung in elektronischen Systemen.

Der Verlag schützt seine ebooks vor Missbrauch des Urheberrechts durch ein digitales Rechtemanagement. Bei Kauf im Webshop des Verlages werden die ebooks mit einem nicht sichtbaren digitalen Wasserzeichen individuell pro Nutzer signiert.

Bei Kauf in anderen ebook-Webshops erfolgt die Signatur durch die Shopbetreiber. Angaben zu diesem DRM finden Sie auf den Seiten der jeweiligen Anbieter.

*Für
Janne, Julia, Daniel und Katrin*

Einleitung

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 Programmierkenntnisse.

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 beherrscht 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. → **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 Programmerversuche 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.

Regelmäßig 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.

Kapitel 1

Aller Anfang ist schwer? Einstieg in JavaScript

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
- 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

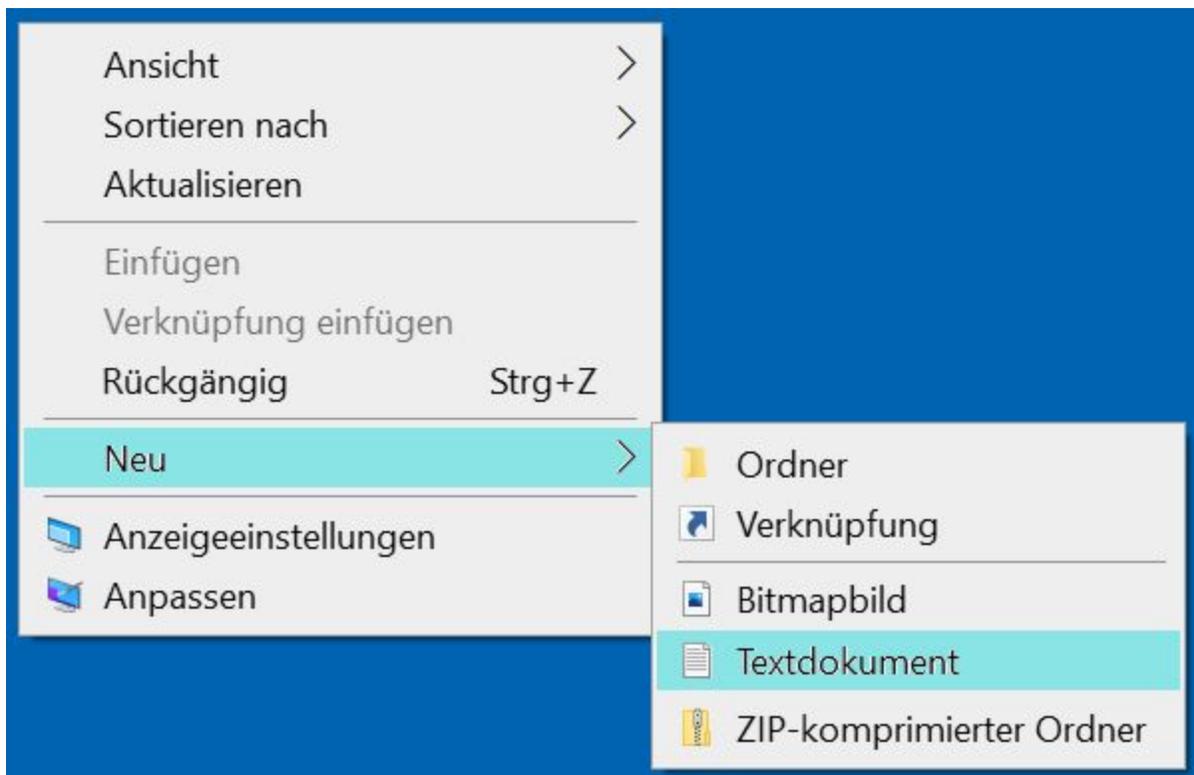
Ein erstes Hallo, schlicht und einfach

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 mitinstalliert 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.

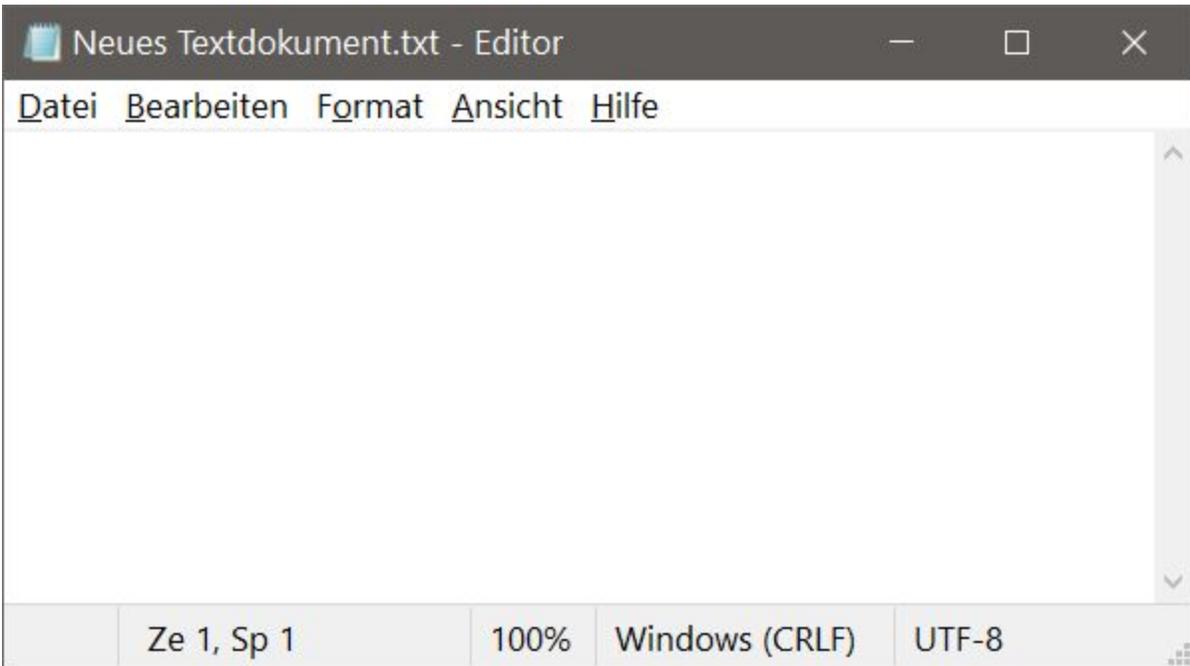
- Klicke mit der rechten Maustaste auf den Desktop, ein Kontextmenü öffnet sich, klicke dort auf **NEU** und dann auf **TEXTDOKUMENT**.



Auf dem Desktop findest du nun ein neues Symbol.

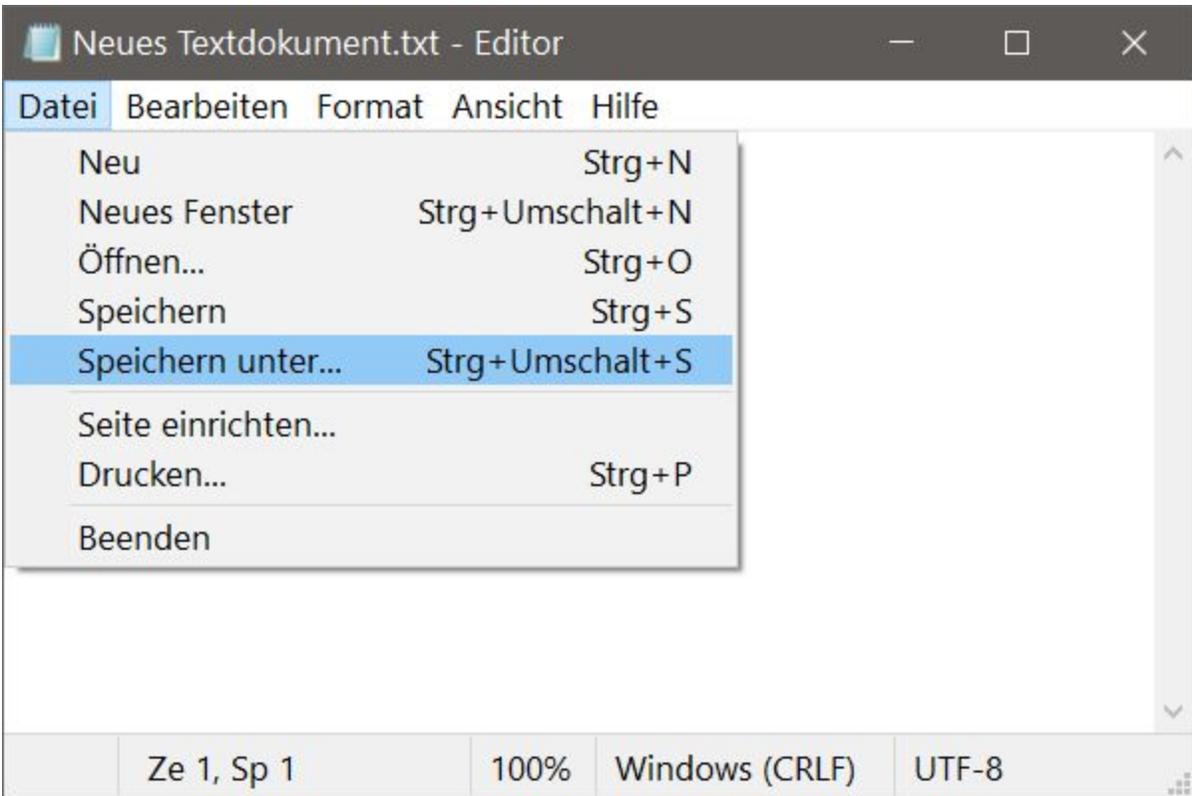


- Wenn du auf dieses Symbol doppelklickst, öffnet sich kurze Zeit später der zugehörige Editor und bietet dir ein leeres Fenster an.

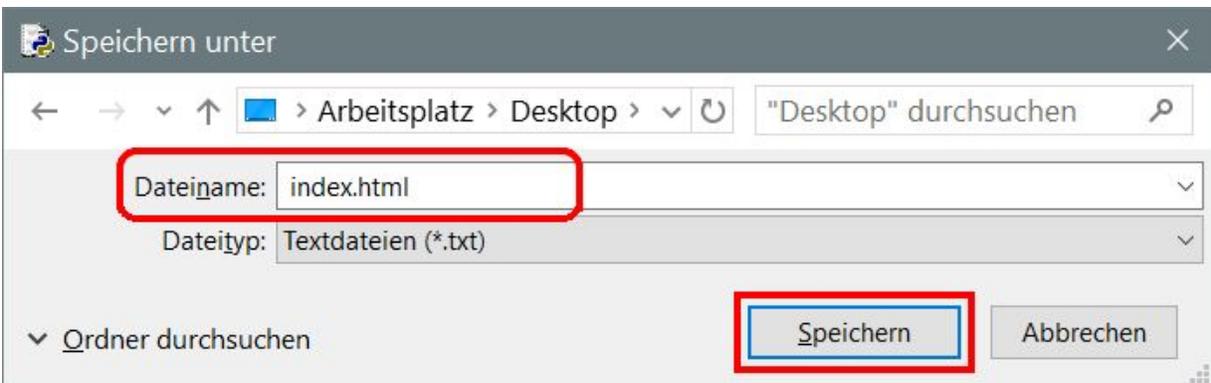


Bevor wir da etwas hineintippen, speichern wir die noch leere Datei unter einem passenden Namen:

- Klicke im Editor oben auf **D**ATEI und dann auf **S**PEICHERN **U**NTER.



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



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

- 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