

Ralph Steyer

# Webanwendungen erstellen mit Vue.js

MVVM-Muster für konventionelle  
und Single-Page-Webseiten

**EBOOK INSIDE**

 Springer Vieweg

---

# Webanwendungen erstellen mit Vue.js

---

Ralph Steyer

# Webanwendungen erstellen mit Vue.js

MVVM-Muster für konventionelle und  
Single-Page-Webseiten

Ralph Steyer  
Bodenheim, Deutschland

ISBN 978-3-658-27169-5      ISBN 978-3-658-27170-1 (eBook)  
<https://doi.org/10.1007/978-3-658-27170-1>

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

Springer Vieweg

© Springer Fachmedien Wiesbaden GmbH, ein Teil von Springer Nature 2019

Das Werk einschließlich aller seiner Teile ist urheberrechtlich geschützt. Jede Verwertung, die nicht ausdrücklich vom Urheberrechtsgesetz zugelassen ist, bedarf der vorherigen Zustimmung des Verlags. Das gilt insbesondere für Vervielfältigungen, Bearbeitungen, Übersetzungen, Mikroverfilmungen und die Einspeicherung und Verarbeitung in elektronischen Systemen.

Die Wiedergabe von allgemein beschreibenden Bezeichnungen, Marken, Unternehmensnamen etc. in diesem Werk bedeutet nicht, dass diese frei durch jedermann benutzt werden dürfen. Die Berechtigung zur Benutzung unterliegt, auch ohne gesonderten Hinweis hierzu, den Regeln des Markenrechts. Die Rechte des jeweiligen Zeicheninhabers sind zu beachten.

Der Verlag, die Autoren und die Herausgeber gehen davon aus, dass die Angaben und Informationen in diesem Werk zum Zeitpunkt der Veröffentlichung vollständig und korrekt sind. Weder der Verlag, noch die Autoren oder die Herausgeber übernehmen, ausdrücklich oder implizit, Gewähr für den Inhalt des Werkes, etwaige Fehler oder Äußerungen. Der Verlag bleibt im Hinblick auf geografische Zuordnungen und Gebietsbezeichnungen in veröffentlichten Karten und Institutionsadressen neutral.

Springer Vieweg ist ein Imprint der eingetragenen Gesellschaft Springer Fachmedien Wiesbaden GmbH und ist ein Teil von Springer Nature.

Die Anschrift der Gesellschaft ist: Abraham-Lincoln-Str. 46, 65189 Wiesbaden, Germany

---

## Vorwort

Selbstverständlich ist das Internet aktuell bzw. immer noch in aller Munde. Digitalisierung allgemein ist ja eines der am meisten benutzten „Buzzwords“ – vor allen Dingen von Politikern, Medienleuten und Entscheidern. Doch die Zeiten von **statischen** Webseiten sind weitgehend vorbei. War es vor wenigen Jahren noch üblich, dass zumindest einfache Webseiten von Privatpersonen oder kleineren Vereinen noch reinen HTML-Code (Hypertext Markup Language) verwendet haben, findet man heutzutage solche antiquierten Webseiten immer weniger. In der Regel kommen auch hier zumindest Content-Management-Systeme (CMS) wie WordPress, Joomla!, Typo oder Drupal zum Einsatz oder solche Webseiten werden zumindest mit Stylesheets und/oder JavaScript aufgepeppt.

Doch wenn man mit diesen 08/15-Lösungen der gängigen CMS nicht zufrieden ist oder diese erst gar nicht den verlangten Anforderungen genügen (können), bleibt nur die echte Programmierung von Webseiten oder gar Webapplikationen. Dieser Weg beginnt auf der einen Seite meist mit Programmierung aufseiten des Webservers samt nachgeschalteten Datenbanksystemen, aber auf der anderen Seite muss auch zwingend bei modernen Webseiten oder Webapplikationen im Client (dem Browser) programmiert werden.

Und im Browser steht seit Jahren zur Programmierung an sich ausschließlich JavaScript als universell verfügbare Technologie bereit. Zusammen mit HTML und CSS (Cascading Style Sheets) bildet JavaScript den Dreiklang der modernen Webseiten und vor allem der clientseitigen Web-Programmierung.

Nun gibt es JavaScript zwar schon sehr lange im Web, wurde aber über die meisten Jahre vollkommen unterschätzt und als primitive Einsteigersprache abqualifiziert. Erst in den letzten Jahren erkennt man, welchen Schatz man mit JavaScript zur effizienten und leistungsfähigen Programmierung zur Verfügung hat und dass diese Sprache alles andere als eine Einsteigersprache ist – auch wenn man sie als Anfänger schnell lernen kann. Ganz im Gegenteil. Nur man muss für eine professionelle Anwendung damit auch umgehen können, denn im Gegensatz zu Rund-um-Sorglos-Programmierwelten aus dem .NET- und Java-Umfeld, die Programmierer über Jahre „verweichlicht“ haben, indem Sie diese kaum noch Fehler machen ließen, kann man in JavaScript solche Schutzmechanismen

nicht erwarten. Effizientes und sicheres Programmieren mit JavaScript fordert Skills vom Programmierer, statt diese IDEs (Integrated Development Environment – Integrierte Entwicklungsumgebung) und Laufzeitumgebungen zu übertragen. Aber damit ist JavaScript auch viel schlanker und effizienter als die mittlerweile hoffnungslos überladenen Konkurrenten. Die professionelle, effiziente und sichere Programmierung mit JavaScript setzt deshalb jedoch eine immense Programmiererfahrung voraus. Schon im Client, aber erst recht auf Serverseite, wo JavaScript mittlerweile ebenfalls seinen Siegeszug angetreten hat. JavaScript ist wie ein Skalpell. In den Händen eines erfahrenen Chirurgen kann man damit Wunderdinge vollführen. In den Händen eines Laien oder einer nicht sorgfältig arbeitenden Person lässt sich damit ein immenser Schaden anrichten. Wobei JavaScript das Spagat hinbekommt, dennoch auch von Einsteigern für einfache Aufgaben leicht verwendet werden zu können. Ich höre in meinen JavaScript-Schulungen immer wieder, dass viele Teilnehmer schon JavaScript verwendet haben. Meist wurden vorhandene Skripte kopiert und ggf. angepasst oder ganz einfache Skripts selbst geschrieben und in HTML-Seiten gebaut. Und das funktioniert meist auch, wobei fast immer die Ergänzung der Teilnehmer war, dass sie nicht so recht wussten, warum es denn geht.

Die vielen Frameworks, die sich im Umfeld des Webs die letzten Jahre etabliert haben, verwenden zwar einige grundsätzlich verschiedene Ansätze. Aber die meisten Frameworks versuchen vielfach JavaScript um Dinge zu erweitern, die damit in der Kernversion so nicht gehen. Damit wird eine Erleichterung im Umgang mit JavaScript geschaffen und zudem werden Möglichkeiten bereitgestellt, die man mit reinem JavaScript nicht oder nicht so einfach zur Verfügung hat.

Dabei gehen die verschiedenen Frameworks vom Ansatz her ganz unterschiedliche Wege, um letztendlich im Code, der beim Anwender landet, doch wieder bei dem vollkommen gleichen Ergebnis zu landen, nämlich ein Konglomerat aus HTML, CSS und JavaScript. Natürlich immer verbunden mit Ressourcen wie Bildern, Videos, Audios und Ähnlichem.

Nun ist Vue.js ein immer populärer werdendes Framework im Web, das einen sehr speziellen Ansatz wählt. Es handelt sich um ein reaktives, clientseitiges JavaScript-Webframework, das man im Wesentlichen zur Erstellung sogenannter Single-Screen-Webanwendungen bzw. Single-Page-Webanwendungen (nur eine Webseite im Browser, die bei Bedarf Teile aktualisiert und keine neue Webseite nachlädt) nach einem Model-View-Controller-Muster (MVC) benutzt. Genau genommen wird nach einem Model-View-View-Controller-Muster (MVVC) gearbeitet. Aber man kann durchaus auch Webapplikationen und Webseiten damit erzeugen, die aus mehreren Seiten aufgebaut sind.

Dabei ist Vue.js sowohl recht einfach zu erlernen als auch sehr gut erweiterbar bzw. anpassbar. Um Vue.js erfolgreich lernen zu können, genügen gute Kenntnisse in HTML und JavaScript sowie wenn möglich CSS, was ich auch in dem Buch voraussetzen möchte. Die Entwickler von Vue.js nennen das Framework „progressiv“. Das bedeutet im Wesentlichen, dass es genauso für kleine Verbesserungen an einzelnen Details der

---

Webseite genutzt werden kann wie für größere Projekte. Darüber hinaus wird die Erstellung von wiederverwendbaren Komponenten unterstützt. Ein weiterer Vorteil von Vue.js ist, dass es keine komplexe Installation verlangt und bei Bedarf sogar gänzlich ohne Installation „aus der Cloud“ (von einem CDN – Content Delivery Network) genutzt werden kann.

Folgen Sie mir also in die faszinierende Welt von Vue.js!

Frühjahr/Sommer 2019

Ihr Autor  
Ralph Steyer

<http://www.rjs.de>

---

# Inhaltsverzeichnis

<b>1</b>	<b>Einleitung – Bevor es richtig losgeht</b>	<b>1</b>
1.1	Was behandeln wir in dem einleitenden Kapitel?	1
1.2	Das Ziel des Buchs.	1
1.3	Was sollten Sie bereits können?	3
1.4	Was benötigen Sie zum Arbeiten mit dem Buch?	3
1.4.1	Das Framework Vue.js	5
1.5	Die Features von Vue.js	10
1.5.1	Direktiven	10
1.5.2	Der virtuelle DOM	10
1.5.3	Die Datenbindung und Reaktivität	11
1.5.4	Erstellung von Komponenten	11
1.5.5	Ein eigenes Eventsystem	12
1.5.6	Animations- und Übergangseffekte	12
1.5.7	Berechnete Eigenschaften	12
1.5.8	Templates/Vorlagen	13
1.5.9	Watcher	13
1.5.10	Routing	13
1.5.11	Vue-CLI	13
<b>2</b>	<b>Erste Beispiele – Vue.js einfach einmal testen</b>	<b>15</b>
2.1	Was behandeln wir in dem Kapitel?	15
2.2	Das Grundgerüst und ein erstes Beispiel	15
2.3	Dynamik für das Beispiel	18
2.3.1	Echte Reaktion und v-model	20
<b>3</b>	<b>Hinter den Kulissen – Wie und warum funktioniert Vue.js?</b>	<b>25</b>
3.1	Was behandeln wir in diesem Kapitel?	25
3.2	Das Prinzip der Fehlertoleranz und das DOM-Konzept	25
3.2.1	Das DOM-Konzept aus einer besonderen Sicht	26

3.3	Arrays, Objekte und JSON	32
3.3.1	Hash-Listen	35
3.3.2	Die JavaScript Object Notation	36
3.3.3	Callbacks und Funktionsreferenzen	37
3.4	MVC und MVVC	39
3.4.1	Entwurfsmuster	41
3.4.2	Das MVC-Pattern	41
3.4.3	MVVC	43
<b>4</b>	<b>Vue.js in der Tiefe – Die Vue-Instanz, Vue-Templates und die Datenbindung</b>	<b>45</b>
4.1	Was behandeln wir in dem Kapitel?	45
4.2	Die <i>Vue</i> -Instanz	46
4.2.1	Auf die Erzeugung des <i>Vue</i> -Objekts reagieren – der Lebenszyklus	47
4.3	Grundsätzliches zu Vue.js-Templates	48
4.3.1	Das <i>template</i> -Attribut	49
4.3.2	Unter der Template-Haube	49
4.3.3	Verschiedene Arten der Datenbindung in Templates	50
4.3.4	Verwenden von JavaScript-Ausdrücken bei Datenbindung	60
4.4	Mehr zu Direktiven	60
4.4.1	Argumente	61
4.4.2	Dynamische Argumente	61
4.4.3	Einschränkungen für dynamische Argumentwerte	61
4.4.4	Modifikatoren bei Attributen	62
4.5	Komponenten	62
4.5.1	Aufpassen!	64
4.5.2	Globale versus lokale Registrierung	64
4.5.3	Datenübergabe	65
4.5.4	Der Weg zurück – Slots	67
4.5.5	Asynchrone Datenübertragung	68
4.5.6	Einzelne Dateikomponenten – Single File Components	69
4.6	Welche Seite hätten Sie denn gerne? Routing	70
4.6.1	MVVC/MVC und das Routing	71
4.6.2	Die konkrete Umsetzung in Vue.js	71
<b>5</b>	<b>Mit Arrays arbeiten – Iterationen mit der v-for-Direktive</b>	<b>75</b>
5.1	Was behandeln wir in dem Kapitel?	75
5.2	Die v-for-Direktive	75
5.2.1	Statische Anzeige von Werten aus einem Array	76
5.2.2	Zugriff auf den Index des aktuellen Elements	78

5.3	Zugriff auf komplexere Strukturen . . . . .	80
5.3.1	Verschachtelte v-for-Direktiven . . . . .	83
5.3.2	Einzelne Einträge direkt ansprechen . . . . .	91
5.4	Besondere Anwendungen der v-for-Direktive . . . . .	91
5.4.1	Die v-for-Direktive mit einem Wertebereich (Range). . . . .	92
5.4.2	Zugriff auf die Elternelemente. . . . .	94
5.4.3	Schlüssel und Index in einem Objekt. . . . .	95
5.4.4	Das key-Attribut zur Bindung der Id . . . . .	95
5.4.5	Callbacks aufrufen . . . . .	96
5.5	Änderungen bei Arrays beobachten . . . . .	98
5.5.1	Mutierende Methoden . . . . .	98
5.5.2	Arrays sortieren und mit Methoden arbeiten . . . . .	100
5.5.3	Neue Arrays generieren . . . . .	106
<b>6</b>	<b>Bedingtes Rendern – Die v-if-Direktive – Entscheidungen treffen . . . . .</b>	<b>109</b>
6.1	Was behandeln wir in dem Kapitel? . . . . .	109
6.2	Die v-if, v-else und v-else-if-Direktiven . . . . .	109
6.3	Die v-show-Direktive . . . . .	113
6.4	Wann v-if und wann v-show? . . . . .	113
6.5	Eine besondere Kombination – die Direktive v-for mit v-if oder v-show. . . . .	114
6.5.1	Ein Wrapper mit v-if ist besser . . . . .	115
<b>7</b>	<b>Ereignisse, Methoden, Beobachter und berechnete Eigenschaften – kalkulierte Ergebnisse und Reaktionen . . . . .</b>	<b>119</b>
7.1	Was behandeln wir in dem Kapitel? . . . . .	119
7.2	Grundsätzliche Überlegungen zur Aufgabenverteilung. . . . .	119
7.3	Methoden eines Vue-Objekts und die methods-Eigenschaft . . . . .	120
7.4	Das Eventhandling in Vue.js . . . . .	121
7.4.1	Hintergründe der Ereignisbehandlung. . . . .	121
7.4.2	Das konkrete Beispiel für v-on . . . . .	124
7.4.3	Das Eventobjekt auswerten . . . . .	127
7.4.4	Event Modifier . . . . .	130
7.4.5	Weitere Modifizierer . . . . .	131
7.4.6	Benutzerdefinierte Ereignisse . . . . .	132
7.5	Die <i>computed</i> -Eigenschaft. . . . .	134
7.6	Wann Methoden und wann berechnete Eigenschaften?. . . . .	137
7.7	Watcher (Beobachter) . . . . .	138
7.7.1	Beobachten der Geolokalisierung mit einem Watcher . . . . .	138
7.7.2	Ajax mit einem Watcher. . . . .	141

<b>8</b>	<b>Dynamische Layouts mit Datenbindung – Stylesheets</b>	
	<b>dynamisieren</b>	149
8.1	Was behandeln wir in dem Kapitel?	149
8.2	Datenbindung und die <i>v-bind</i> -Direktive bei bedingten Klassen	149
8.2.1	Umschalten von CSS-Klassen	150
8.2.2	Die Array-Notation für mehrere Eigenschaften	157
8.2.3	Logik in der HTML-Datei	158
8.3	Datenbindung und die <i>v-bind</i> -Direktive bei Inline-Styles	158
8.4	Abkürzungen (Shorthands)	160
8.4.1	Die <i>v-bind</i> Abkürzung	160
8.4.2	Abkürzung <i>v-on</i>	160
<b>9</b>	<b>Formulare und Formulardatenbindung – Interaktion mit dem Anwender</b>	161
9.1	Was behandeln wir in dem Kapitel?	161
9.2	Grundlagen der Formularverwendung im Web	162
9.2.1	Die enthaltenen Formularelemente	162
9.3	Grundlegende Verwendung der Formulardatenbindung in Vue.js	163
9.3.1	Vue-Instanz first	163
9.4	Einige konkrete Beispiele	164
9.4.1	Ein einfaches Formular mit verschiedenen Formularelementen	164
9.5	Dynamische Optionen	170
9.6	Eine Aufgabenliste als praktisches Beispiel	172
9.6.1	Eine erste einfache Version einer Todo-List	172
9.6.2	Eine dauerhafte Aufgabenliste	176
9.6.3	Persistenz die Zweite – serverseitig	181
9.7	Mehr zur Wertebindungen bei Formularen	187
9.7.1	Die Modifikatoren	188
<b>10</b>	<b>Filtertechniken – Nur ausgewählte Daten</b>	189
10.1	Was behandeln wir in dem Kapitel?	189
10.2	Grundlagen von Filtern bei JavaScript-Arrays	189
10.2.1	Die Pfeilnotation	192
10.3	Filter bei Vue.js	192
10.3.1	Lokale Filter	193
10.3.2	Globale Filter durch Erweiterung der Vue-Instanz	193
10.3.3	Dynamische Filterung	194
10.3.4	Filter verketteten	196
10.3.5	Übergabe an Parameter	196

---

<b>11</b>	<b>Übergänge und Animationen – Bewegende Dinge</b> .....	197
11.1	Was behandeln wir in dem Kapitel? .....	197
11.2	Übergänge mit transition .....	198
11.3	Die Übergangsklassen .....	201
11.4	CSS-Animationen .....	202
11.5	Spezielle Situationen .....	203
11.5.1	Übergänge und Animationen zusammen verwenden .....	203
11.5.2	Explizite Übergangszeiten – die duration-Angabe .....	203
11.5.3	JavaScript-Hooks .....	203
11.5.4	Animation von Daten .....	205
<b>12</b>	<b>Ausblicke – Was gibt es noch in Vue.js?</b> .....	207
12.1	Was behandeln wir in dem Kapitel? .....	207
12.2	Vue.js in CMS oder in Kombination mit anderen Frameworks verwenden .....	207
12.3	Serverseitiges Rendern .....	209
12.4	Mixins .....	210
12.5	Benutzerdefinierte Direktiven .....	210
12.6	Plugins .....	211
12.6.1	Ein Plugin verwenden .....	212
12.6.2	Ein Plugin schreiben .....	212
	<b>Anhang</b> .....	215
	<b>Stichwortverzeichnis</b> .....	219

---

## Über den Autor



**Ralph Steyer** ist Diplom-Mathematiker und arbeitet als freiberuflicher Trainer, Autor und Programmierer. Unter <http://www.rjs.de> finden Sie seine Webseite und unter <http://blog.rjs.de> seinen Blog. Seine beruflichen Schwerpunkte liegen in der Webentwicklung sowie Programmierung in Java und .NET.

Hier ist noch ein kurzer Abstract der beruflichen Laufbahn und Erfahrungen:

- Studium bis 1990 in Frankfurt/Main an der Johann Wolfgang Goethe-Universität.
- Nach dem Studium Programmierer bei einer großen Versicherung in Wiesbaden für versicherungsmathematische PC-Programme.
- Nach knapp 4 Jahren innerbetrieblicher Wechsel in die Konzeption von Großrechnerdatenbanken.
- Seit 1996 Freelancer. Aufteilung der Arbeit in verschiedene Tätigkeitsgebiete – Fachautor, Fachjournalist, EDV-Dozent und Programmierer/Consultant.
- Zahlreiche Buchpublikationen, Videoproduktionen und Onlinetrainings im IT-Bereich sowie Fachbeiträge in Computermagazinen.
- Speaker auf verschiedenen IT-Konferenzen.
- Lehrbeauftragter an der Hochschule Rhein-Main in Wiesbaden und der TH Bingen.

---

## 1.1 Was behandeln wir in dem einleitenden Kapitel?

Bevor es richtig losgeht, sollen in diesem einleitenden Kapitel einige Dinge geklärt werden, die Ihnen die folgende Arbeit mit diesem Buch und Vue.js erleichtern werden. Insbesondere erfahren Sie, was Sie als Voraussetzungen mitbringen sollten und woher Sie Vue.js bekommen. Und es wird kurz besprochen, was Vue.js an Features mitbringt.

---

## 1.2 Das Ziel des Buchs

Dieses Buch ist zum **Einstieg** in die Verwendung des Frameworks Vue.js konzipiert. Entweder in Form des Selbststudiums oder als Begleitmaterial in entsprechenden Kursen. Vermittelt werden die elementaren Grundlagen, um Webapplikationen mit Vue.js erstellen als auch pflegen zu können. Das umfasst Themen wie die folgenden:

- Das Umfeld – HTML/JavaScript/CSS und das Web
- Grundsätzliche Erstellung von Anwendungen mit Vue.js
- Die JavaScript-Basis – insbesondere Arrays und JSON, Funktionsreferenzen sowie das DOM-Konzept
- Das MVVC-Konzept als spezielle Variante des MVC-Konzepts
- Die *Vue*-Instanz und wie man damit arbeitet
- Ereignisbehandlung
- Watcher
- Berechnete Eigenschaften
- Komponenten und deren Lebenszyklus

- Die Double Curly Syntax und Datenbindung
- Direktiven
- Templates
- Modularität und die Erweiterung von Vue.js

Bei dem Vue.js-Framework kommt im Buch die Version 2.x (Stand Anfang 2019) zum Einsatz.

### ► Definition

Der Begriff „**Framework**“ ist in seiner Form nicht so ganz eindeutig. Laut Wikipedia versteht man darunter das:

Ein Framework (englisch für Rahmenstruktur) ist ein Programmiergerüst, das in der Softwaretechnik, insbesondere im Rahmen der objektorientierten Softwareentwicklung sowie bei komponentenbasierten Entwicklungsansätzen, verwendet wird. Im allgemeineren Sinne bezeichnet man mit Framework auch einen Ordnungsrahmen.

Allgemein versteht man darunter, dass ein Framework eine reale oder konzeptuelle Struktur ist, die als Unterstützung oder Anleitung konzipiert wurde, um etwas zu erstellen, welches die Struktur selbst zu etwas Sinnvollem erweitert. In der IT versteht man darunter oftmals eine mehrschichtige Struktur, die angibt, welche Arten von Programmen erstellt werden können beziehungsweise sollen und wie sie miteinander zusammenhängen. Einige Frameworks beinhalten Programme, spezifizieren Programmierschnittstellen (APIs -Application Programming Interface) oder bieten Programmier-Werkzeuge, die mit dem Framework genutzt werden können. In der Regel wird man in der IT bei einem Framework mindestens eine Bibliothek samt einem Regelwerk für deren Anwendung haben. Manchmal gibt es auch eine noch weiter gefasste Form, bei der neben gewissen Bibliotheken und Syntaxstrukturen beziehungsweise Sprachen explizit Tools wie Visual Studio und/oder SQL-Server in das Konzept voll integriert werden.

Bei den Unterlagen wird Wert auf die grundsätzliche Anwendung der verschiedenen Techniken und einfache Beispiele gelegt, weniger auf Vollständigkeit aller möglichen Anweisungen, Befehle oder Parameter. Insbesondere soll und kann hier auch nur ein Einstieg die Erstellung einer Vue.js-Applikation verschafft werden. Bei diversen weiterführenden Themen wird jedoch auf die offizielle Dokumentation oder zusätzliche Quellen verwiesen, um Ihnen auch da einen Einstieg zu eröffnen.

- Die Quellcodes des Buchs finden Sie nach Kapiteln und darin erstellten Projekten sortiert auf den Webseiten des Verlags. Die Namen der jeweilig aktuellen Dateien bzw. Projekte werden als Hinweise oder direkt im Text vor den jeweiligen Beispielen angegeben und bei Bedarf wiederholt. Ich empfehle allerdings, dass Sie die Beispiele unbedingt alle von Hand selbst erstellen. Das ist für das Verständnis und das Lernen eindeutig besser als ein reines Kopieren oder nur ansehen.

An einigen Stellen im Buch werden Aufgaben formuliert, die Sie an der Stelle lösen sollten. Bei einigen wenigen Aufgaben (etwa dem Erstellen eines bestimmten Programms) wird explizit auf die Lösung im Anhang verwiesen, wenn es notwendig ist und die Ausführungen an der Stelle die Lösung der Aufgabe nicht weiter erklären oder beschreiben.

---

### 1.3 Was sollten Sie bereits können?

Dieses Buch ist als Unterstützung in einen Einstieg in Vue.js konzipiert, der die Grundlagen dieses Frameworks von Grund auf erarbeitet. Allerdings wird in die Arbeit mit so einem Framework sehr selten ohne Vorkenntnisse im Web und/oder in einer Programmiersprache eingestiegen. Deshalb sollen gute Kenntnisse in HTML und Grundlagen in CSS vorausgesetzt werden. Und dann bleibt noch JavaScript. Für das Verständnis der Buchs sollen die grundlegenden syntaktischen Grundlagen (Einbindung in Webseiten, Datentypen samt der losen Typisierung, Variablen, Schleifen, Entscheidungsstrukturen, Sprungweisungen, einfache Funktionsdeklarationen samt dem Aufruf von Funktionen etc.) ebenso vorausgesetzt werden.

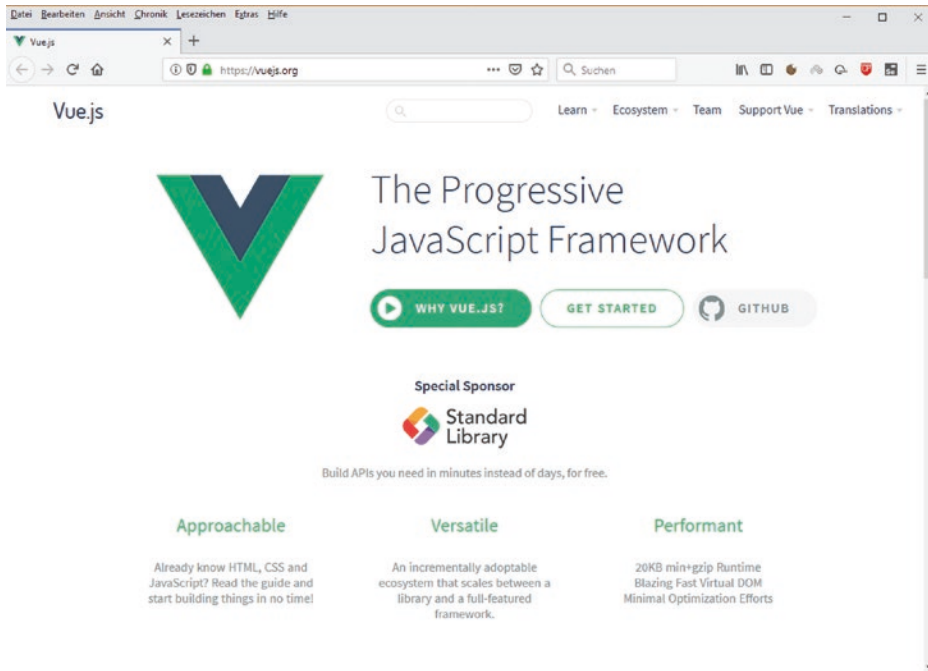
Die wirklich interessanten und nicht ganz trivialen Dinge in JavaScript sind jedoch die Funktionsreferenzen bzw. die Callback-Philosophie und die (weitgehende) Äquivalenz von Objekten und Arrays sowie insbesondere das JSON-Format (JavaScript Object Notation). Zusammen mit dem DOM-Konzept (Document Object Model) sind das die absoluten Grundlagen, um die Arbeitsweise von Vue.js zu verstehen. Zwar kann man auch mit Vue.js auch „arbeiten“, ohne in den Techniken sattelfest zu sein. Ich möchte jedoch so weit gehen und behaupten, dass sich Vue.js fast von selbst erschließt, wenn man ein wenig die Denkweise des MVVC-Musters und eben diese drei Schlüsseltechnologien wirklich gut verstanden hat. Aus diesem Grund werden in dem Buch diese Konzepte auch gleich zu Beginn erst einmal gründlich erläutert.

- ▶ Das Framework Vue.js selbst samt der offiziellen Dokumentation zu dem Framework finden Sie im Web (Abb. 1.1) unter <https://vuejs.org/> bzw. <https://vuejs.org/v2/guide/>. Bei diversen weiterführenden Fragen und Themen wird wie gesagt immer wieder darauf verwiesen.

---

### 1.4 Was benötigen Sie zum Arbeiten mit dem Buch?

- Als Basis für das Buch wird ein PC o. ä. vorausgesetzt.
- Das Referenzbetriebssystem ist Windows 10 (auch Vorgängerversionen wie Windows 7 sind möglich, werden aber nicht explizit berücksichtigt), aber Sie können – wie üblich in der Webprogrammierung – auch mit anderen Betriebssystemen wie Linux oder MacOS arbeiten.



**Abb. 1.1** Die offizielle Webseite von Vue.js

- Darüber hinaus genügt ein beliebiger Editor, aber Sie können auch eine IDE wie Visual Studio oder Visual Studio Code, Eclipse etc. verwenden.
- Ansonsten ist ein lokaler Webserver sehr zu empfehlen. Wenn Sie sich die Sache möglichst einfach machen wollen, bietet sich ein Rundum-Sorglos-Paket wie XAMPP an, wobei sich gerade unter Windows auch die Internet Information Services (IIS) eignen.

Das XAMPP-Paket bezeichnet eine Sammlung von Programmen mit dem Webserver Apache im Zentrum, der durch das Datenbankmanagementsystem MySQL bzw. in neuen Versionen dessen Fork MariaDB (inklusive phpMyAdmin zur Administration des Datenbankmanagementsystems) und PHP-Unterstützung, den FTP-Server FileZilla sowie einige weitere Webtechnologien ergänzt wird. XAMPP steht für verschiedene Betriebssysteme zur Verfügung (<http://www.apachefriends.org/de/>).

Sie brauchen dieses Paket mit einem einfachen Assistenten nur zu installieren und schon haben Sie einen voll funktionstüchtigen Apache-Webserver in einer Grundkonfiguration zur Verfügung. Beachten Sie aber, dass XAMPP in der Grundeinstellung ausschließlich für lokale Testzwecke konfiguriert ist. Um die Sache möglichst einfach zu halten, sind sämtliche Sicherheitseinstellungen niedrig

eingestellt. Sobald die Installation von XAMPP fertig ist, können Sie entweder Apache manuell starten oder aber auch so einrichten, dass Apache als Dienst beziehungsweise Prozess in Ihrem Betriebssystem integriert und sogar automatisch beim Start des Rechners aufgerufen werden kann. XAMPP stellt zur Verwaltung ein komfortables und sehr einfach anzuwendendes Kontrollprogramm zur Verfügung.

### 1.4.1 Das Framework Vue.js

Natürlich brauchen Sie dann noch Vue.js selbst. Auf der Webseite des Projekts gibt es bereits mehrere gute Hinweise, wie man mit der Arbeit anfangen kann (Abb. 1.2).

Vor allen Dingen finden Sie dort Hinweise, wie Sie Vue.js konkret in Ihrer Webseite verwenden können. Das geht bemerkenswert einfach und ist eines der Highlights dieses Frameworks.

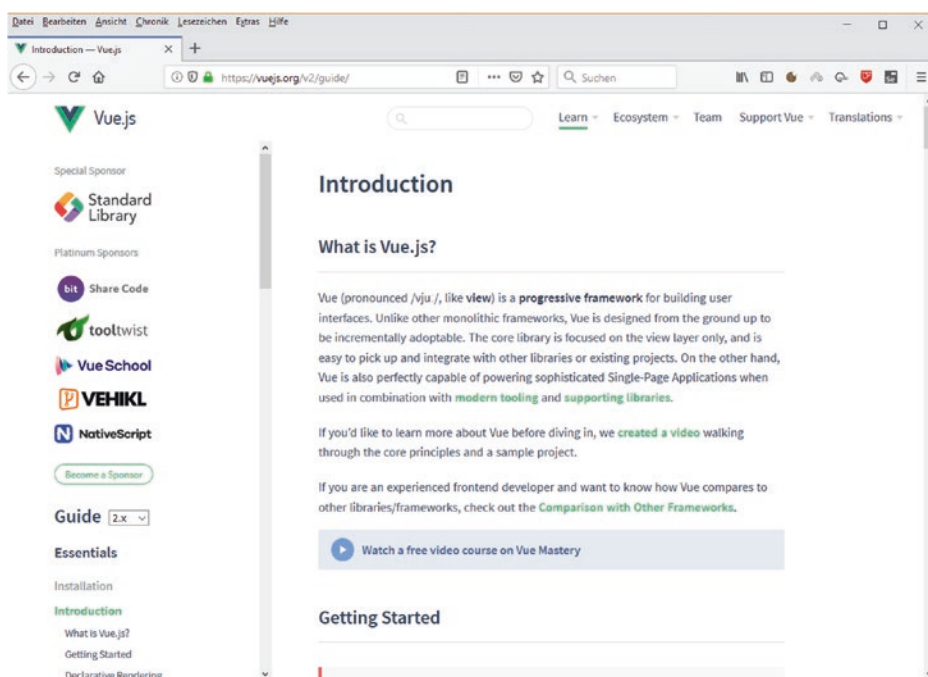


Abb. 1.2 Informationen zu Vue.js

- ▶ Unter <https://vuejs.org/v2/guide/join.html> können Sie dem Team rund um Vue.js selbst beitreten (Join the Vue.js Community!).

Da es sich bei Vue.js im Kern nur um eine JavaScript-Bibliothek handelt, binden Sie das Framework im Grunde wie jede gewöhnliche externe JavaScript-Datei ein. Dabei können Sie – auch einfaches JavaScript – diese Datei von einem fremden oder auch dem eigenen Webserver einbinden.

Beachten Sie – es gibt Vue.js in zwei Varianten:

- Eine minimierte Version für den Praxiseinsatz.
- Eine nicht minimierte Version für die Entwicklung. Der Code dieser Variante ist besser zu lesen und bietet Warnungen bei gängigen Fehlern. Deshalb eignet sie sich zur Entwicklungszeit besser und beim Veröffentlichen tauschen Sie einfach den Link aus.

### 1.4.1.1 Einbinden von einem CDN

Insbesondere wird Vue.js von dem Projekt bereits auf diese Weise zur Einbindung bereitgestellt. Sie brauchen nur den auf der Webseite des Projekts angegebenen Link im

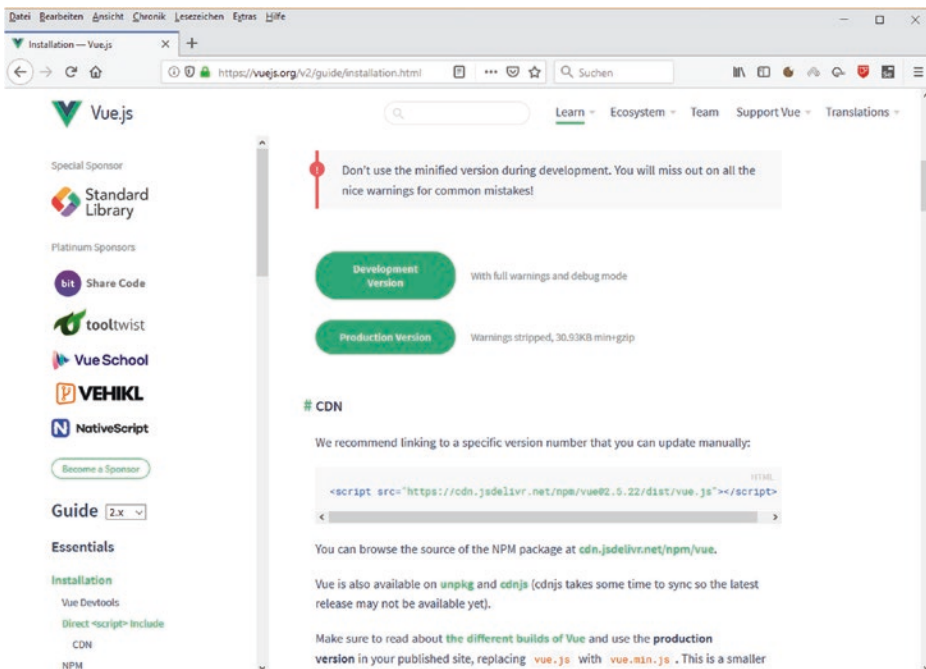


Abb. 1.3 Vue.js einbinden

*Script*-Tag angeben. Dieser lautet etwa <https://cdn.jsdelivr.net/npm/vue/dist/vue.js>, aber das kann sich natürlich verändern und die genauen Details finden Sie jeweils auf der Webseite des Vue.js-Projekts (Abb. 1.3).

In dem Zusammenhang wird von einem von CDNgesehen, was aber im Grunde nur die Art der Bereitstellung, aber nicht die konkrete Einbindung betrifft.

So könnte das etwa im Kopfbereich der HTML-Datei erfolgen:

```
<!DOCTYPE html>
<html>
<head>
...
<script src="https://cdn.jsdelivr.net/npm/vue/dist/vue.js" type="text/
javascript" ></script>
...
</head>
<body>
...
</body>
</html >
```

Beachten Sie, dass in HTML5 die Angabe des MIME-Typs (Multipurpose Internet Mail Extensions) *“text/javascript”* nicht mehr nötig ist und auch nicht gemacht werden soll. Ich widerspreche dieser Vorgabe explizit, denn die Angabe ist auf der einen Seite ohne irgendwelche negativen Auswirkungen und verdeutlicht, dass man ausdrücklich mit JavaScript arbeitet – einige ältere Browser brauchen das.

### 1.4.1.2 Das Framework über eigene Webserver bereitstellen

Sie können Vue.js aber auch über den eigenen Webserver bereitstellen. Dazu laden Sie die JavaScript-Datei des Frameworks einfach über den Browser und den Download-Link auf der Webseite des Projekts und stellen diese auf Ihrem Webserver bereit. Das sieht dann etwa so aus, wenn Sie die übliche Verzeichnisstruktur *lib/js* für Ihre JavaScripts einhalten wollen:

```
<!DOCTYPE html>
<html>
<head>
...
<script src="lib/js/vue.js" type="text/javascript" ></script>
...
</head>
<body>
...
</body>
</html >
```

### 1.4.1.3 Weitere Einbindungsmöglichkeiten

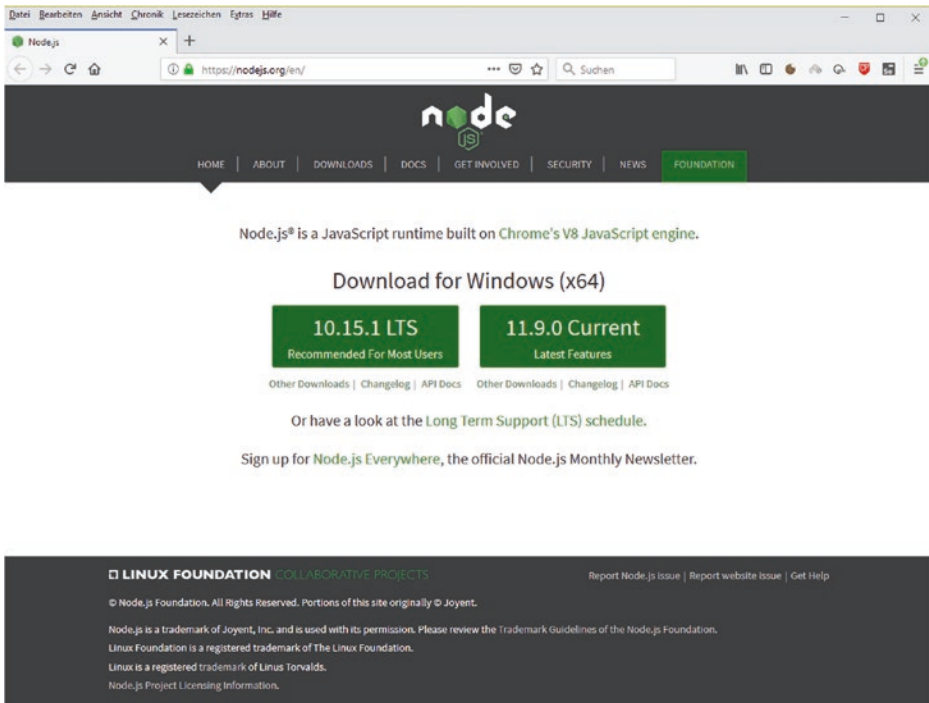
Es gibt nun noch ein paar weitere Einbindungsmöglichkeiten für Vue.js (bower, npm etc.), für die aber auf die Dokumentation (<https://vuejs.org/v2/guide/installation.html>) verwiesen sei und die meines Erachtens nur in wenigen Ausnahmefällen explizite Vorteile bieten.

Wie immer bei Webseiten gilt, dass externe Stylesheets vor den externen JavaScript-Dateien eingebunden werden müssen. Und Skriptdateien eines Frameworks werden **immer vor** den eigenen JavaScript-Dateien eingebunden – sonst werden Ihre Vue.js-Anweisungen nicht funktionieren.

#### ► **Tipp**

Obwohl es für die Arbeit mit JavaScript und Vue.js nicht zwingend erforderlich ist, ist die Installation eines sogenannten **Paketmanagers (Packagemanager)** oft hilfreich. Viele Projekte – gerade im Open-Source-Bereich – setzen mittlerweile auf solche Paketmanager, um ihre Ressourcen bereitzustellen. Eine solche Softwarepaket-Verwaltung (englisch package management software) ermöglicht die komfortable Verwaltung von Software, die irgendwo in Programmpaketform vorliegt. Dazu gehören das Installieren, Aktualisieren und Deinstallieren der Ressourcen. Dahinter liegen in der Regel **Repositories** (verwaltete Verzeichnisse zur Speicherung und Beschreibung von digitalen Ressourcen – meist samt einer Versionierung), in denen diese Ressourcen direkt über das Internet bereitgestellt werden. Sie können dann über die jeweiligen Paketmanager die gewünschten Ressourcen automatisiert installieren, entfernen, erweitern oder auch aktualisieren. Dabei werden vor allen Dingen spezifische Details der Plattformen sowie Abhängigkeiten von anderen Ressourcen beachtet und automatisch angepasst.

Im JavaScript-Umfeld hat sich nun der Paketmanager **npm** (ehemals die Abkürzung für Node Package Manager, aber mittlerweile als rekursives Akronym für „npm is not an acronym“ verwendet – ein recht üblicher Gag in der Open-Source-Szene) ziemlich weit verbreitet. Das ist ein Paketmanager für die JavaScript-Laufzeitumgebung **Node.js** (<https://nodejs.org/en/>), der automatisch mit dieser installiert wird, wenn Sie das bei der Installation nicht abwählen. Node.js selbst ist eine serverseitige, ressourcensparende JavaScript-Plattform, deren Installation ich empfehlen möchte – auch wenn wir in dem Buch nicht weiter darauf eingehen. Damit lassen sich auf Basis von JavaScript alle möglichen Netzwerkanwendungen realisieren – von einem Webserver über Socketserver bis hin zu Pushservices. Und Sie erhalten wie gesagt den npm-Paketmanager als Dreingabe und diesen können Sie dann auch



**Abb. 1.4** Informationen und Ressourcen zu Node.js

für die verschiedensten anderen Ressourcen verwenden. Auf der Webseite von Node.js (Abb. 1.4) finden Sie verschiedene Installationspakete samt den passenden Anleitungen und Dokumentationen.

#### ► **Tipp**

Vue.js ist explizit als JavaScript-Framework konzipiert. Allerdings wird auch das Zusammenspiel mit TypeScript unterstützt. Ganz vereinfacht gesagt ist das ein Ableger von JavaScript mit expliziten Datentypen und strenger Typisierung, der von Microsoft entwickelt wurde. Wobei bei genauerem Hinsehen viel mehr Features dort bereitgestellt werden, die diverse Vorschläge zum zukünftigen ECMAScript 6-Standard bereits vorwegnehmen. So gibt es etwa echte Klassen und Vererbung, Interfaces und echte Namensräume. Aber da die gängigen Browser nur JavaScript statt TypeScript verstehen, muss jeder TypeScript-Code mit einem TypeScript-Compiler vor einem Einsatz in der Praxis in einen JavaScript-Code nach ECMAScript 3 (ES3) oder ECMAScript 5 (ES5) kompiliert werden.

Umgekehrt ist jeder JavaScript-Code auch gültiger TypeScript-Code. Wenn Sie TypeScript im Zusammenhang mit Vue.js explizit einsetzen wollen, müs-

sen Sie ein paar Dinge beachten. Das wird im Buch nicht weiter verfolgt, aber in der Dokumentation von Vue.js finden Sie dazu Hinweise (<https://vuejs.org/v2/guide/typescript.html#ad>).

---

## 1.5 Die Features von Vue.js

Vue.js ist ein sehr leichtgewichtiges Framework und gilt auch von der Leistung als sehr schnell. Vue.js bezeichnet sich selbst als „Progressive JavaScript Framework“. Progressiv bedeutet in dem Zusammenhang, dass Sie Vue.js bei Bedarf auch nur für einen Teil Ihrer Anwendung nutzen können, die dynamischer/interaktiver sein soll. Aber Vue.js stellt noch eine ganze Reihe an speziellen weiteren Features bereit.

### 1.5.1 Direktiven

Vue.js verfügt über integrierte **Direktiven** (Richtlinien), die in dem Framework alle mit dem Präfix *v-* beginnen. Beispielsweise *v-for*, *v-if*, *v-else*, *v-show*, *v-on*, *v-slot*, *v-bind* oder *v-model*. Auf deren Basis werden die verschiedenen Aktionen im Frontend (der View, also der HTML-Seite) ausgeführt. Etwa kann man damit Elemente verbinden und überwachen, wenn sich deren Inhalte ändern (Datenbindung), Iterationen über Elemente ausführen, Entscheidungen treffen, Elemente anzeigen oder wegblenden etc. Man kann sogar eigene Direktiven für das Framework erstellen.

### 1.5.2 Der virtuelle DOM

Allgemein gelten direkte Änderungen am DOM bei der Webprogrammierung als ziemlich schwierig (wegen Browserabhängigkeiten und Synchronisationsproblemen – nicht wegen der Syntax) als auch schlecht für die Performance einer Webseite. Diverse Tipps zur Optimierung von JavaScript drehen sich darum, Aktionen auf dem DOM möglichst zusammenzufassen, um das Rendern des DOMs so selten wie möglich zu erzwingen.

Noch weiter gehen Frameworks wie Vue.js, aber auch andere verwandte Frameworks wie React oder Ember. Sie verwenden einen sogenannten virtuellen DOM. Eventuelle Änderungen werden damit nicht direkt am DOM vorgenommen, sondern es wird mit einer Kopie des DOM gearbeitet. Dieser wird in Form von internen JavaScript-Datenstrukturen erstellt. Wir werden im Laufe des Buchs sehen, dass diese Hashlisten immens effizient sind.

Wenn von dem Framework Änderungen vorgenommen werden sollen, werden diese erst einmal an den JavaScript-Datenstrukturen vorgenommen und diese wird dann mit der ursprünglichen Datenstruktur (dem eigentlichen DOM) verglichen. Mehrere Ände-

rungen können so zusammengefasst werden (etwa das Erstellen eines neuen Elements und das Erstellen und Hinzufügen von Textinhalten zu diesem Element) und dann erst auf den realen DOM übertragen werden. Und auch sonst sind verschiedene Optimierungen im Hintergrund möglich.

### 1.5.3 Die Datenbindung und Reaktivität

Allgemein kann mit der sogenannten „Datenbindung“ eine unmittelbare und direkte Beziehung zwischen zwei Variablen, beispielsweise Ausdrücken, erstellen. Damit assoziieren Sie den Wert eines Ziels mit dem Wert eines gebundenen Ausdrucks. So ein gebundener Ausdruck kann ein einfacher Wert eines beliebigen Typs, ein Objekt, das Ergebnis einer Funktion oder eines Ausdrucks sein. Datenbindung ist Teil des Eventhandlings und beileibe nicht auf Frameworks wie Vue.js und verwandte Frameworks wie Angular.js oder React beschränkt. Das Konzept findet man in mächtigen Programmiersprachen wie C, C++, C# oder Java. Nur spricht man dort oft von Pointern, Referenzen oder Verweisen, die auf eine andere Variable verweisen, genaugenommen den Speicherbereich der Variablen. Ändert sich die Variable dort, ändert sich auch der Wert, der über den Pointer verfügbar ist. Das ist eigentlich so eingängig, dass es teils schon fast trivial erscheint.

Das Verfahren gibt es aber auch in „primitiven“ Anwendungsprogrammen wie Excel oder Libre Calc. Dort gibt es Referenzbezüge zwischen Zellen in einer Tabelle, wenn diese in Formeln in anderen Zellen verwendet werden. Wenn sich der Wert der referenzierten Zelle auf irgendeine Weise ändert, werden sich alle anderen daran gebundenen Ausdrücke in der Weise ändern, wie die Bindungsvorschrift (die Formel) das verlangt. Und das ohne Verzögerung und vor allen Dingen ohne dass die Aktualisierung noch manuell ausgelöst werden muss.

Das Konzept der Datenbindung ist die Kernfunktion des Frameworks. Diese bidirektionale Datenbindung hilft immens bei der Bearbeitung oder Zuweisung von Werten zu HTML-Attributen, dem Ändern des CCS-Stils oder dem Zuordnen von Klassen. Insbesondere die Bindungsdirektive *v-bind* macht die die Arbeit sehr komfortabel, aber auch andere Direktiven wie *v-model* spielen hier eine Rolle.

Vue.js wird deshalb auch als reaktiv (reactive) bezeichnet, denn in dem Moment, wo sich die Daten in der Anwendung ändern, kümmert sich Vue.js automatisch darum, dass sie überall dort, wo sie auf der Webseite benutzt werden, ebenfalls verändert werden.

### 1.5.4 Erstellung von Komponenten

Für den Siegeszug der objektorientierten Programmierung ist die Wiederverwendbarkeit das ausschlaggebende Argument gewesen. Sogenannte Komponenten sind eine der wichtigsten Funktionalitäten von Vue.js. Mit deren Hilfe können benutzerdefinierte Elemente erstellt werden, die in HTML wiederverwendet und im Model angesprochen