

## 9 Single-File-Component (SFC) mit Composition-API

Dieses Kapitel stellt die modernen Single-File-Components (SFC) in einer .vue-Datei mit Composition-API und Setup-Direktive (`<script setup>`) vor.

Zitat aus der Dokumentation: "<script setup> is a compile-time syntactic sugar for using Composition-API inside Single-File-Components (SFCs)." [<https://v3.vuejs.org/api/sfc-script-setup.html>]. Die Syntax `<script setup>` gilt seit Version 3.2 als stabil [<https://blog.vuejs.org/posts/vue-3.2.html>].

### 9.1 Aufbau einer Single-File-Component (SFC)

Das folgende Schaubild veranschaulicht den Aufbau einer Single-File-Components (SFC) in einer .vue-Datei mit Composition-API und Setup-Direktive (`<script setup>`).

#### Single-File-Component (.vue-Datei) mit Composition-API und Setup-Direktive

##### Vorlagenbereich

`<template>`

- Interpolation mit `{{ x }}`
- Attributbindung mit `v-bind:name="Wert"`
- Ereignisbehandlungen mit `@ereignisname`, z.B. `@click`
- Unterkomponenten per Tagename, z.B. `<Abc propName="Wert"/>`

##### Vorlagenbezogenes CSS (Scoped Styles)

`<style scoped>`

##### Codebereich

`<script setup lang="ts">`

- Imports, z.B. `import { ref } from 'vue';`
- Parameter mit `defineProps()`
- Watcher mit `watch()`
- Ereignisse mit `defineEmits()`
- Öffentliche Methoden mit `defineExpose()`
- Reaktive Variablen mit `ref()` oder `reactive()`
- Normale, nicht-reaktive Variablen
- Lebenszyklusereignisbehandlung, z.B. `onMounted()`
- DOM-Ereignisbehandlung, z.B. für `@click`
- Hilfsfunktionen

## 9.2 SFC-Beispiel

Im Kontrast zum vorherigen Kapitel zeigt das folgende Listing den Quadratwurzelrechner nun als wiederverwendbare Single-File-Component (SFC) in einer .vue-Datei mit Composition-API und Setup-Direktive (`<script setup>`).

Die wichtigsten Unterschiede zur Implementierung mit dem Options-API sind:

- Die Programmiersprache ist TypeScript (`lang="ts"`).
- Die SFC enthält keine vollständige Webseite, sondern nur ein Fragment.
- Das Fragment steht in `<template>` (mit der gleichen Template-Syntax wie im vorherigen Kapitel). Dank TypeScript entfällt aber die Typkonvertierung beim Ausdruck `'Nächster Wert: ' + (x+1)`.
- Die CSS-Deklarationen sind mit `<style scoped>` auf die aktuelle Komponente begrenzt (Scoped Styles).
- Es wird hier keine Vue-Instanz erzeugt, da die Komponente Teil einer Vue.js-Anwendung werden soll. Die Vue-Instanz wird zentral für alle Komponenten erzeugt (dazu später mehr).
- Benötigte Typen von Vue.js müssen importiert werden.
- Variablen, berechnete Werte, Funktionen und Lebenszyklus-Ereignisse stehen im `<script setup>`-Tag nach logischen Gesichtspunkten sortiert.
- Anders als im vorherigen Kapitel kann man in diese wiederverwendbare Komponente einen Parameter mit Namen `startValue` hereinreichen. Die in Vue.js eingebaute Funktion `defineProps()` definiert die öffentlichen Parameter der Komponente. Syntaktisch ist dies ein Makro, das vom Vue.js-Compiler zur Entwicklungszeit übersetzt wird.
- Alle Variablen, die sich verändern können, müssen mit der Hilfsfunktion `ref()` in ein Objekt des Typs `RefImpl` verpackt und damit zu reaktiven Variablen gemacht werden.
- Berechnete Werte müssen mit `computed()` in ein Objekt des Typs `ComputedRefImpl` verpackt werden.
- Beim lesenden und schreibenden Zugriff auf die `RefImpl`-Objekte, muss der Entwickler immer `.value` aufrufen. Wenn man `.value` vergisst, meckert der TypeScript-Compiler: "Type T is not assignable to Ref<T>". Im Template ist die Verwendung von `.value` freilich nicht notwendig: dort werden die Objekte automatisch ausgepackt ("Unwrapping").

Zu beachten ist, dass der Inhalt von `<script setup>` bisher nicht in eine separate Datei ausgelagert werden kann im Sinne einer Template-Datei und einer Code-Behind-Datei ("`<script setup>` cannot be used with the `src` attribute" [<https://v3.vuejs.org/api/sfc-script-setup.html#typescript-only-features>]).

*Listing: Quadratwurzelrechner.vue*

```
<!-- Vue.js Template -->
<template>
  <div id="Quadratwurzelrechner">
    <h2>{{name }}</h2>
    <h4>{{author}}</h4>
    <div>Zahl: <input type="number" v-model="x" style="width:50px;">
      <button @click="increment" :title="'Nächster Wert: ' +
(x+1)">+1</button> Quadratwurzel: {{result.toFixed(2)}}
    </div>
  </div>
</div>
```

```
<h3 v-if="all.size > 0">{{all.size}} Ergebnisse:</h3>
<ul>
  <li v-for="[key,value] in all"
    v-bind:key="key"
    v-bind:class="{ 'even': key%2==0, 'odd': key%2!=0}">
    <!-- or: v-bind:style="{color: (key%2==0 ? 'green' :
'blue'))}"> -->
    {{key}} = {{value.toFixed(2)}}
  </li>
</ul>
</div>
</template>

<!-- Scoped CSS -->
<style scoped>
  li { margin: 2px; }

  li::marker { color:red; }
  .even { color: green}
  .odd { color: blue}
</style>

<!-- Composition-API in Single-File-Components (SFCs) -->
<script setup lang="ts">
import { ref, computed, onMounted } from 'vue';
import { version } from 'vue';

// Parameter
const props = defineProps({
  startValue: Number
});

// Variablen
const name = "Quadratwurzelrechner in Vue.js " + version;
let author = ref('(C) Dr. Holger Schwichtenberg');
let x = ref(props.startValue || 0);
let all = new Map();

// Berechnete Werte
let result = computed(() => Math.sqrt(x.value));

// Benutzerereignis
function increment()
{
  x.value++;
  all.set(x.value, result.value);
}
```

```
// Lebenszyklusereignis
onMounted(() => {
  console.log(name + ": mounted!");
  author.value += " " + new Date().getFullYear();
});
</script>
```

Das Schlüsselwort `this` ist in solchen Single-File-Components übrigens nicht belegt. Wir befinden uns auch offensichtlich nicht in einer TypeScript-Klasse, denn wir müssen Variablen mit `let` bzw. `const` deklarieren und das Schlüsselwort `function` vor alle Funktionen schreiben. Eine Kombination von `<script setup>` mit Vue Class Component ist noch nicht möglich (siehe [<https://github.com/vuejs/vue-class-component/issues/416>]).

## 9.3 Nutzung einer Single-File-Component

Die Single-File-Component aus dem vorherigen Unterkapitel kann man in der vorliegenden Form nicht einfach in einem Browser ausführen, denn sie braucht als Umgebung ein Vue.js-Projekt mit Startcode, Layout-Seite, TypeScript-Transpiler und Bundler.

Zum einfachen, isolierten Test einer Komponente gibt es eine solche Umgebung auf einer vom Vue.js-Team betriebenen Website, dem "Vue SFC Playground" [<https://sfc.vuejs.org>]. Hier kann man den obigen Code erfassen und dann direkt im Browserfenster ausführen, siehe folgende Abbildung.

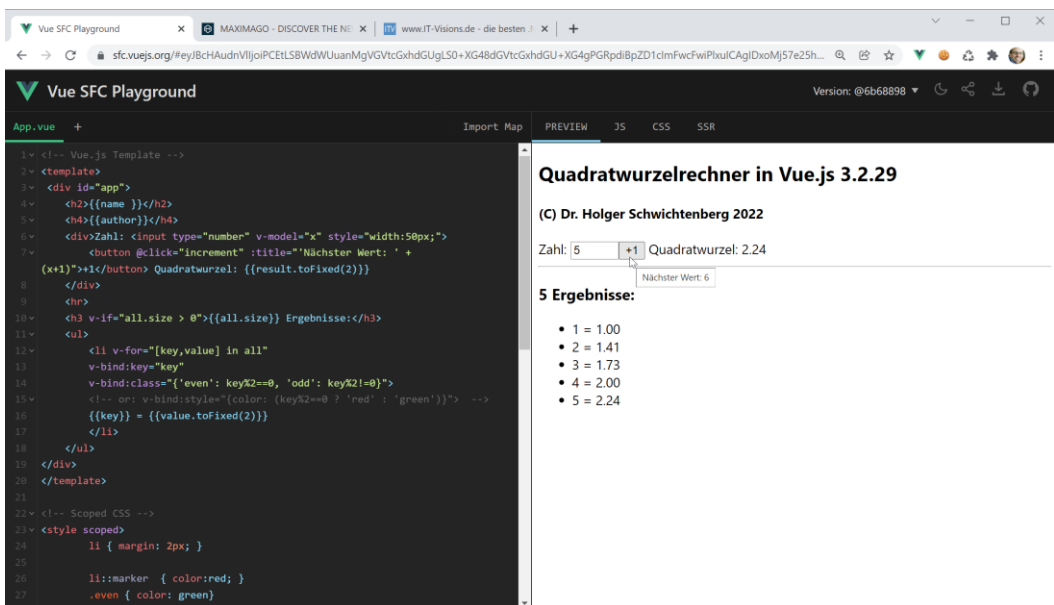


Abbildung: Die Vue.js-Komponente aus dem Listing im Vue SFC Playground

Für die echte Softwareentwicklung braucht man ein Vue.js-Projekt. Solch ein Projekt erschafft man besser nicht von Hand, sondern über die Vue CLI [<https://cli.vuejs.org>] oder das Werkzeug Vite [<https://vitejs.dev>]. Hier soll die etablierte Vue CLI zum Einsatz kommen.

Die zuvor gezeigte Single-File-Component `Quadratwurzelrechner.vue` kann man als eigenständige Webseite in eine Vue.js-Anwendung einbetten oder über den Komponentennamen ohne `.vue` als Tagnamen:

```
<Quadratwurzelrechner :startvalue="10"></Quadratwurzelrechner>
```

Der Doppelpunkt vor dem Parameter `startValue` zeigt an, dass der Inhalt nicht – wie in HTML-Attributen üblich – als einfache Zeichenkette, sondern per Vue.js-Datenbindung erfolgen soll; hier könnte also ein beliebiger JavaScript-Ausdruck folgen. Die Wertübergabe ohne den Doppelpunkt würde in diesem Fall zwar auch funktionieren, aber Vue.js würde sich in der Browserkonsole leicht beschweren: "[Vue warn]: Invalid prop: type check failed for prop "startvalue". Expected Number with value 10, got String with value "10"."

# 10 Vue.js-Projekte mit der Vue CLI

## 10.1 Anlegen des Projekts mit der Vue CLI

Für den weiteren Verlauf dieses Buchs sollten die im Kapitel 6 dokumentierten Voraussetzungen (Node.js, Visual Studio Code, Volar) installiert haben.

Dann installieren Sie die Vue CLI wie folgt per Node Package Manager:

```
npm install -g @vue/cli@4.5.15
```

Etwas unbehaglich wird einem Entwickler, weil bei der Installation der Vue CLI zahlreiche Warnungen bezüglich nicht mehr gepflegter Pakete und Sicherheitslücken erscheinen, siehe folgende Abbildung. Aber solche Missstände sind aufgrund der Kurzlebigkeit vieler NPM-Pakete in der Webwelt leider in vielen Frameworks so üblich. Ebenso die Tatsache, dass man sehr viele NPM-Pakete braucht: Wie man in der folgenden Abbildung sieht, hat die Vue CLI stolze 942 NPM-Abhängigkeiten.

Als ersten Test nach dem Installieren prüft man die Versionsnummer der Vue CLI:

```
vue --version
```

```
PS T:\> npm install -g @vue/cli
npm WARN deprecated @hapi/topo@3.1.6: This version has been deprecated and is no longer supported or maintained
npm WARN deprecated @hapi/bourne@1.3.2: This version has been deprecated and is no longer supported or maintained
npm WARN deprecated urix@0.1.0: Please see https://github.com/lydell/urix#deprecated
npm WARN deprecated har-validator@5.1.5: this library is no longer supported
npm WARN deprecated resolve-url@0.2.1: https://github.com/lydell/resolve-url#deprecated
npm WARN deprecated apollo-tracing@0.15.0: The 'apollo-tracing' package is no longer part of Apollo Server 3. See https://www.apollographql.com/docs/apollo-server/migration/#tracing for details
npm WARN deprecated graphql-extensions@0.15.0: The 'graphql-extensions' API has been removed from Apollo Server 3. Use the plugin API instead: https://www.apollographql.com/docs/apollo-server/integrations/plugins/
npm WARN deprecated @hapi/address@2.1.4: Moved to 'npm install @sideway/address'
npm WARN deprecated uid@3.4.0: Please upgrade to version 7 or higher. Older versions may use Math.random() in certain circumstances, which is known to be problematic. See https://v8.dev/blog/math-random for details.
npm WARN deprecated uid@3.4.0: Please upgrade to version 7 or higher. Older versions may use Math.random() in certain circumstances, which is known to be problematic. See https://v8.dev/blog/math-random for details.
npm WARN deprecated apollo-cache-control@0.14.0: The functionality provided by the 'apollo-cache-control' package is built in to 'apollo-server-core' starting with Apollo Server 3. See https://www.apollographql.com/docs/apollo-server/migration/#cachecontrol for details.
npm WARN deprecated request@2.88.2: request has been deprecated, see https://github.com/request/request/issues/3142
npm WARN deprecated @hapi/hoek@8.5.1: This version has been deprecated and is no longer supported or maintained
npm WARN deprecated @hapi/joi@15.1.1: Switch to 'npm install joi'
npm WARN deprecated graphql-tools@4.0.8: This package has been deprecated and now it only exports makeExecutableSchema.\nAnd it will no longer receive updates.\nWe recommend you to migrate to scoped packages such as @graphql-tools/schema, @graphql-tools/utils and etc.\nCheck out https://www.graphql-tools.com to learn what package you should use instead

changed 941 packages, and audited 942 packages in 34s

67 packages are looking for funding
  run 'npm fund' for details

12 vulnerabilities (4 moderate, 8 high)

Some issues need review, and may require choosing
a different dependency.

Run 'npm audit' for details.
PS T:\> C
PS T:\> vue --version
@vue/cli 4.5.15
PS T:\> |
```

Abbildung: Warnungen beim Installieren der Vue CLI

**Hinweis:** Zum Zeitpunkt der ersten Erstellung dieses Textes (Ende Januar 2022) war die Version 4.5.15 der Vue CLI aktuell; diese Version ist am 28. Oktober 2021 erschienen.

Danach, am 17.5.2022, ist zwar die Version 5.0 der Vue CLI erschienen [<https://github.com/vuejs/vue-cli/releases>]. Allerdings zeigte sich in verschiedenen Tests, dass diese noch sehr fehlerhaft ist. So kann diese Version der Vue CLI noch mal erfolgreich ein Grundprojekt mit Nightwatch-Tests für Chrome erstellen, siehe Screenshot und <https://github.com/vuejs/vue-cli/issues/4689>

Diese Version der Vue CLI ist daher leider noch nicht praxisreif und das Buch bleibt daher vorerst auf der Version 4.5.15.

Wenn Sie dennoch es mit der Vue CLI Version > 5 versuchen wollen, nutzen Sie den obigen Befehl ohne Angabe einer Versionsnummer, um die aktuellste Version zu installieren:

```
npm install -g @vue/cli
```

Dieses Buch wird auf die Version 5.0 umgestellt, sobald die Vue CLI 5 auch mit Chrome korrekt funktioniert.

```
PowerShell
Vue CLI v5.0.1
+ Creating project in T:\VueCLI5\miraclelist-client.
+ Initializing git repository...
+ Installing CLI plugins. This might take a while...

added 1410 packages, and audited 1411 packages in 27s

136 packages are looking for funding
  run 'npm fund' for details

7 moderate severity vulnerabilities

To address all issues (including breaking changes), run:
  npm audit fix --force

Run 'npm audit' for details.
+ Invoking generators...
+ Installing additional dependencies...

npm ERR! code ETARGET
npm ERR! notarget No matching version found for chromedriver@99.
npm ERR! notarget In most cases you or one of your dependencies are requesting
npm ERR! notarget a package version that doesn't exist.

npm ERR! A complete log of this run can be found in:
npm ERR!   C:\Users\hs\AppData\Local\npm-cache\_logs\2022-03-03T08_53_22_654Z-debug.log
ERROR Error: command failed: npm install --loglevel error --legacy-peer-deps
Error: command failed: npm install --loglevel error --legacy-peer-deps
  at ChildProcess.<anonymous> (C:\Pakete\npm\node_modules\@vue\cli\lib\util\executeCommand.js:138:16)
  at ChildProcess.emit (node:events:376:20)
  at ChildProcess.cp.emit (C:\Pakete\npm\node_modules\@vue\cli\node_modules\cross-spawn\lib\enoent.js:34:29)
  at maybeClose (node:internal/child_process:1063:16)
  at Process.ChildProcess._handle.onexit (node:internal/child_process:295:5)
PS T:\VueCLI5>
```

Abbildung: Die Vue CLI Version 5.0.1 scheitert schon beim Anlegen eines neuen Projekts, weil die Nightwatch-Tests eine Beta-Version des Chrome-Driver (99) verwenden, die noch gar nicht in NPM verfügbar ist

Search packages

Search

chromedriver

98.0.1 • Public • Published 17 days ago

Readme

Explore BETA

7 Dependencies

464 Dependents

136 Versions

Tip: Click on a version number to view a previous version's package page

Version	Downloads (Last 7 Days)	Tag
98.0.1	126.833	latest

Version History

Version	Downloads (Last 7 Days)	Published
98.0.1	126.833	17 days ago
98.0.0	35.403	24 days ago
97.0.4	30.193	a month ago
97.0.3	1.097	a month ago
97.0.2	17.670	a month ago
97.0.1	1.207	a month ago

Install

> npm i chromedriver

Repository

github.com/giggio/node-chromedriver

Homepage

github.com/giggio/node-chromedriver

Weekly Downloads

642.892

Version

98.0.1

License

Apache-2.0

Unpacked Size

37.2 kB

Total Files

6

Issues

7

Pull Requests

1

Abbildung: Liste der derzeit verfügbaren Versionen des Chrome-Driver auf NPM