



# Entwickeln im Sprint

## Effiziente Webentwicklung mit Visual Studio Code

**Visual Studio Code gehört zu den beliebtesten kostenlosen Entwicklungsumgebungen mit einem großen Angebot an Erweiterungen. Mit wenig Einrichtungsarbeit wird der Editor auf allen Betriebssystemen zur Arbeitsumgebung für effiziente Webentwicklung und schnelle Prototypen mit HTML und CSS.**

**Von Manuel Ottlik**

**M**anchmal muss es einfach schnell gehen: Wer mit dem Entwickeln eines Web-Projekts beginnt, einen Prototyp abliefern muss, eine neue Funktion ausprobieren oder einen c't-Programmier-

artikel nachvollziehen will, möchte sich nicht lange mit dem Einrichten einer Arbeitsumgebung aufhalten. Ein lokaler Webserver muss her, für die meisten CSS-Projekte ist ein Präprozessor wünschenswert – und der Editor sollte etwas mehr können als nur Code-Highlighting.

Eine Option ist, viele Tools für diese Aufgaben zu installieren und einzurichten. Wesentlich schneller zum Ziel kommt man mit Visual Studio Code (VSC oder VS Code), aufgebohrt mit den richtigen Erweiterungen. Die IDE gibt es kostenlos zum Download für Linux, macOS und Windows (siehe [ct.de/yxur](http://ct.de/yxur)). Entwickelt wird VSC unter Open-Source-Lizenz mit großem Personaleinsatz von Microsoft-Mitarbeitern.

### Projektstruktur

Der Artikel verwendet als anschauliches Beispiel für die schnelle Webentwicklung

eine einfache statische Steckbrief-Webseite. Neben einer Willkommenseite soll es eine „Über Mich“-Seite und ein Impressum geben – natürlich mit ausgelagerten Styles. Legen Sie für das Projekt einen leeren Ordner an und öffnen Sie diesen mit „File/Add folder to workspace“ in Visual Studio Code. Die Arbeit beginnt in der Datei `index.html`. Wenn Sie das Beispiel nicht abtippen möchten, finden Sie das fertige Projekt auf GitHub über [ct.de/yxur](http://ct.de/yxur) zum Download.

Die Index-Datei könnten Sie jetzt mit dem kompletten Standardaufbau für HTML-Do5-Dokumente versehen – entweder aus dem Gedächtnis oder irgendwo aus dem Internet herauskopiert. Beides ist nicht gerade elegant oder effizient. Besser macht es das Werkzeug Emmet, das in VSC eingebaut und für HTML-Dokumente aktiviert ist. Emmet macht es dem Nutzer leicht, mit wenigen Tastaturanschlä-

gen große HTML-Strukturen aufzubauen – mit einer Syntax, die an die der CSS-Selektoren angelehnt ist. Statt das Grundgerüst für HTML-Dokumente selbst zu schreiben, können Sie einfach die Abkürzung ! schreiben. Mit Betätigen der Tabulator- oder Enter-Taste wird das Kürzel durch HTML ersetzt:

```
<!-- HTML-Grundgerüst in Emmet: ! -->
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="UTF-8">
<meta name="viewport" content="width=device-width, initial-scale=1.0">
<meta http-equiv="X-UA-Compatible" content="ie=edge">
<title>Document</title>
</head>
<body>
</body>
</html>
```

Die Emmet-Abkürzungen für sämtliche HTML-Elemente (zum Beispiel `<div>`, `<table>` oder `<img>`) sind leicht zu merken: Lassen Sie einfach die spitzen Klammern weg und bestätigen Sie mit der Tab- oder Enter-Taste. Visual Studio Code zeigt erkannte Emmet-Abkürzungen beim Tippen bereits in einem Tooltip an. Die Emmet-Abkürzungen sind aber nur aktiv, wenn Sie diese eintippen – nicht beim Einfügen aus der Zwischenablage.

Mit `div` entsteht ein komplettes `div`-Element, also `<div></div>`. Um einem Tag direkt eine Klasse mitzugeben, hängen Sie diese einfach mit einem Punkt an, eine ID erzeugen Sie mit `#`. Die Abkürzung `div.klasse#test` erzeugt also Folgendes:

```
<div class="klasse" id="test"></div>
```

Auch das lässt sich noch abkürzen. Da ein `div`-Element immer als Standardelement angenommen wird, können Sie `div` auch weglassen und stattdessen `.klasse#test` schreiben. Ganz schön flott, oder?

Jetzt soll die Seite ein paar Elemente im Body-Bereich bekommen: Ein `<div>`-Element mit der Klasse `.container`, in dem die Kind-Elemente `<header>`, `<nav>`, `<section>` und `<footer>` enthalten sind. Um Kinder- und Geschwister in Emmet zu erzeugen, müssen Sie wieder nur die Syntax von CSS-Selektoren kennen. Dort wird ein Kind mit `>` angegeben, ein Geschwisterelement mit einem `+`. Die Abkürzung `.container>header+nav+section+footer` generiert also alles am Stück:

```
<div class="container">
  <header></header>
  <nav></nav>
  <section></section>
  <footer></footer>
</div>
```

VSC setzt außerdem automatisch Tabstopps, also Stellen, an denen Sie noch Inhalt nachfüllen müssen – springen Sie einfach mit der Tab-Taste durch die Elemente und tragen Sie Ihre Inhalte ein. Sie können den Inhalt eines Tags aber auch Emmet direkt mitgeben, indem Sie diesen in geschweifte Klammern hüllen. So erzeugt `h1{Max Mustermann}` eine ausgefüllte Überschrift der ersten Ebene.

## Wiederholt gespart

Ein weiteres nützliches Feature, das viel Tipparbeit spart, ist das Multiplizieren von Elementen. Für das Menü sollen drei `<a>`-Tags mit der Klasse `.nav-element` angelegt werden. Setzen Sie einfach `*3` dahinter, um drei gleiche Elemente zu erzeugen. Die Abkürzung `a.nav-element*3` erzeugt nun drei Hyperlinks, die Sie mit Tabstopps direkt befüllen können. Mit dem `$`-Zeichen greifen Sie innerhalb einer Multiplikation auf einen Zähler zurück. Das kann man zum Beispiel nutzen, wenn man IDs mit einer laufenden Nummer versehen muss. Um zum Beispiel die IDs `item1`, `item2` und `item3` an die Links anzuhängen, reicht: `a.nav-element#item$*3`.

Eine weitere syntaktische Feinheit sind Gruppierungen, die Sie mit `( )` erzeugen. Eine Gruppe definiert, was auf der gleichen Ebene des HTML-Baums liegt. Den Rahmen für den Steckbrief erhalten Sie mit:

```
.container>(header>h2)+j
↳(nav>a.nav-element*3)+section+footer
```

Das Ergebnis sehen Sie im Kasten auf Seite 158. Wenn Sie mit den hier beschriebenen Grundfunktionen vertraut sind und sämtliche Tricks von Emmet lernen wol-

Emmet zeigt in Tooltips an, dass es eine Kurzform erkannt hat. Mit Enter oder Tab wird daraus HTML-Code erzeugt.

len, lohnt ein Blick in die Dokumentation, zu finden über [ct.de/yxur](http://ct.de/yxur).

## Live Server

Womöglich haben Sie während der Arbeit am HTML-Gerüst geschaut, wie Ihre Seite inzwischen aussieht. Im einfachsten Fall kann man eine HTML-Datei direkt im Browser öffnen. Dieses Verfahren ist für kompliziertere Projekte aber nicht geeignet – schon das Einbinden von Bildern, Skripten und Stilen kann je nach Browser Ärger bereiten, wenn der Browser Dateien aus dem lokalen Dateisystem öffnet (erkennbar an `file://` in der Adressleiste).

Stattdessen brauchen Sie einen kleinen lokalen Webserver. Auch einen solchen können Sie bequem in VSC einbinden. Weit verbreitet ist die Erweiterung „Live Server“ von Ritwick Dey. Diese installieren Sie entweder über die Suche im Erweiterungen-Menü (Menüpunkt mit den Blöcken links im Menü) oder über den Marketplace (siehe [ct.de/yxur](http://ct.de/yxur)).

Nach der Installation erscheint unten in der blauen Statusleiste ein neuer Button mit der Beschriftung „Go Live“. Einmal draufgeklickt, startet im Hintergrund ein lokaler Server, schnappt sich einen noch freien Port und öffnet einen neuen Tab in Ihrem Standardbrowser.

Dort wird der Inhalt Ihres Verzeichnisses angezeigt. Haben Sie gerade eine HTML-Datei im Editor geöffnet, wird für diese direkt „Hot Reloading“ aktiviert und der Browser lädt automatisch neu, sobald Sie eine Änderung speichern.

## Besseres CSS

Nachdem das Gerüst steht und sich im Browser bewundern lässt, kann die Arbeit am Design beginnen – ebenfalls mit technischer Unterstützung bei der Fleißarbeit.

Viel Tipperei bleibt Ihnen erspart, wenn Sie statt CSS das Layout in SCSS anlegen. Das ist eine entwicklerfreundlichere Sprache, die Konstrukte versteht, die es

```
test.html > HTML > head
1 <!DOCTYPE html>
2 <html lang="en">
3 <head>
4   <meta charset="UTF-8">
5   <meta name="viewport" content="width=device-width, initial-scale=1.0">
6   <meta http-equiv="X-UA-Compatible" content="ie=edge">
7   link:css
8   <title>D Link:css Emmet Abbreviation
9 </head>
10 <body>
11
12 </body>
13 </html>
```

```
<!DOCTYPE html>
<html lang="en">
<head>
  <meta charset="UTF-8">
  <meta name="viewport"
content="width=device-width,
initial-scale=1.0">
  <meta http-equiv="X-UA-
Compatible" content="ie=edge">
  <link rel="stylesheet"
href="style.css">
  <title></title>
</head>
<body>
  <div class="container">
    <header>
      <h2></h2>
    </header>
    <nav>
      <a href="" class="nav-
element"></a>
      <a href="" class="nav-
element"></a>
      <a href="" class="nav-
element"></a>
    </nav>
  <section></section>
  <footer></footer>
</div>
</body>
</html>
```

Mit nur 66 Tastaturanschlägen entsteht dank Emmet ein umfangreiches HTML-Gerüst.

noch nicht in den CSS-Standard geschafft haben. Für Entwickler am hilfreichsten ist die Möglichkeit, Stile ineinander zu verschachteln. Eine Einführung haben wir bereits veröffentlicht [1]. Den Artikel finden Sie kostenlos über ct.de/yxur.

Damit aus SCSS reines CSS wird, das die Browser verstehen, braucht man einen Compiler, der sich alle SCSS-Dateien schnappt und browsertauglich übersetzt. Eine solche Software kann man theoretisch als separates Programm betreiben. Oder man installiert eine Erweiterung: Populär ist hier „Live Sass Compiler“ vom Entwickler des „Live Server“. Die Erweiterung finden Sie wieder über die Erweiterungs-Suche oder über den Marketplace.

Einmal installiert, nistet sich die Erweiterung ebenfalls unten in der Statusleiste von VS Code ein. Mit dem Button

„Watch Sass“ starten Sie die Überwachung des Arbeitsverzeichnisses. Liegt dort eine Datei mit der Endung .scss oder .sass, erzeugt die Erweiterung sofort eine zugehörige CSS-Datei. Bei jeder Änderung an der SCSS-Quelldatei wird sofort neu übersetzt.

Um SCSS im Beispielprojekt zu nutzen, legen Sie die Datei style.scss an und erstellen Sie im Header-Bereich Ihres HTML-Gerüsts das Tag <link rel="stylesheet" href="style.css" />. Die Emmet-Abkürzung link:css spart mal wieder Arbeit.

### CSS verdichten

Wer mag, kann die Erweiterung mit ein paar Einstellungen anpassen. So lässt sich beispielsweise in der Komprimierung zwischen expanded, also der normalen Ansicht, compact, einer Art Zwischenstufe, und compressed wählen. Da Sie die CSS-Datei ohnehin nicht mehr per Hand bearbeiten, empfiehlt sich die Einstellung compressed. Auf einer fertigen Seite spart diese Einstellung Traffic. Guter Stil ist es, komprimiertes CSS mit der Dateierdung .min.css zu kennzeichnen.

Diese Einstellungen könnten Sie über die Einstellungen (zu erreichen mit „Strg+,“) global für Ihren Nutzeraccount setzen. Sinnvoller ist es aber, die Einstellung direkt im Projektordner abzulegen. Wenn Sie oder ein Kollege diesen Ordner dann – zum Beispiel per Git – auf einem anderen Computer laden, arbeitet der SCSS-Compiler direkt mit den richtigen Einstellungen. Um die Kompression zu aktivieren, legen Sie den Ordner „.vscode“ im Projektverzeichnis an und erzeugen Sie darin die Datei settings.json. Legen Sie hier folgendes Schnipsel ab:

```
{
  "liveSassCompile.settings.formats":
  [
    {
      "format": "compressed",
      "extensionName": ".min.css",
      "savePath": null
    }
  ]
}
```

Die vier Blöcke in der linken Seitenleiste öffnen die Suche nach Erweiterungen: Die Community hat tausende Erweiterungen bereitgestellt.

Da sich der Dateiname der CSS-Datei geändert hat, müssen Sie diese Änderung jetzt noch im HTML nachtragen. Anschließend sollten Sie der Seite etwas Gestaltung spendieren und die Erweiterungen in Aktion testen. So könnten Sie den Container mit einer festen Breite versehen und auf der Seite zentrieren. Auch bei CSS kann Emmet helfen: Für viele CSS-Attribute gibt es Kürzel. Für width zum Beispiel w. Das für die Zentrierung nötige Attribut margin: auto; erzeugen Sie mit m:a. Experimentieren Sie ein wenig mit den Shortcuts. Nebenbei dürfen Sie Ihren Browser beim automatischen Neuladen bewundern – SCSS-Compiler und Server arbeiten hier beim Speichern Hand in Hand.

### Arbeitsfähig

Wer verschiedene Generationen von Web-Entwicklungswerkzeugen miterlebt hat, wird vielleicht erstaunt sein, wie elegant man die Probleme mit einem modernen Werkzeug wie Visual Studio lösen kann. Statt mehrerer externer Tools und aufgeblasener IDEs, die ein halbes Betriebssystem enthalten, kann man mit ein paar Erweiterungen, die fast keine Konfiguration benötigen, viel Zeit sparen. Die Vorteile liegen auf der Hand: Alle Erweiterungen können zentral aus der Entwicklungsumgebung bedient werden – außerdem landen die Einstellungen zusammen mit dem Code auf allen Arbeitsmaschinen.

Unter anderem wegen seiner Erweiterbarkeit hat es VS Code in wenigen Jahren zu großer Beliebtheit gebracht. Der Marketplace für Erweiterungen ist dementsprechend groß und schnelllebig: Es lohnt sich also, ab und an mal im Marketplace nach neuen Erweiterungen Ausschau zu halten, die einem das Leben als Entwickler vereinfachen. (jam@ct.de) ct

### Literatur

- [1] Herbert Braun, Variabel gestylt, Wie neue CSS-Techniken Webdesign vereinfachen, c't 24/2014, S. 210

### Beispielprojekt: ct.de/yxur

