

## Entwicklung eines HTML-Editors unter didaktischen Gesichtspunkten

Dominik Heun<sup>1</sup>

**Abstract:** Im Kerncurriculum Informatik des Landes Hessen für die gymnasiale Oberstufe wird die Umsetzung eines HTML-Projekts gefordert. Dabei stellt sich regelmäßig die Frage, mit welchen Werkzeugen und Editoren eine HTML-Seite geschrieben werden soll. Viele frei verfügbare Editoren stellen weitreichende Funktionen zur Verfügung, die in vielen Fällen zu weit gehen oder Informatikdidaktisch interessante Aspekte außen vorlassen.

In diesem Paper werden didaktische Anforderungen an einen Editor formuliert, die anschließend mit einem selbst entwickelten Editor erfüllt werden sollen. Es werden Grundfunktionen des Editors (HTML, CSS, JavaScript, Dokumentenbaum) vorgestellt, Hilfen in der Erstellung von HTML-Dokumenten diskutiert und Erfahrungen aus dem Einsatz im Unterricht der Einführungsphase vorgestellt.

**Keywords:** Gymnasium, Oberstufe, Einführungsphase, HTML, CSS, Webseiten

### 1 Einleitung

Innerhalb der Einführungsphase (zwei Jahre vor dem Abitur) können Schülerinnen und Schüler an Gymnasien in Hessen das Fach Informatik wählen. Darin erhalten sie nach dem Kerncurriculum der gymnasialen Oberstufe<sup>2</sup> eine informatische Grundbildung in verschiedenen Themenbereichen, die im ersten Halbjahr um das Thema Internet aufgebaut ist. Nach der Behandlung des Themenbereichs Internetprotokolle wird ein HTML-Protokoll angesetzt. Für das HTML-Projekt werden drei grundlegende Inhalte beschrieben: die Hypertext Markup Language selbst (HTML), die Struktur von HTML-Dokumenten sowie Cascading Style Sheets (CSS).

### 2 Kriterien für einen HTML-Editor

Für die Lehrenden ist zu Beginn der Unterrichtsreihe die Auswahl eines geeigneten Editors nötig, der von den Schülerinnen und Schülern genutzt werden soll. Diese Suche gestaltet sich schwierig, da einige Editoren wie bspw. die Standard-Editoren von Windows oder Mac reine Texteditoren sind und keine Unterstützung in Bezug auf HTML-Dokumente geben. Phase 5<sup>3</sup> bietet genauso wie Notepad++<sup>4</sup> eine rudimentäre

---

<sup>1</sup> Justus-Liebig-Schule Darmstadt, Julius-Reiber-Straße 3, 64293 Darmstadt, heun@lio-darmstadt.de

<sup>2</sup> [Hes15]

<sup>3</sup> Für Schulen und Privatanwender kostenlos unter <http://www.phase5.info> (nur Windows)

<sup>4</sup> Kostenlos verfügbar unter <https://notepad-plus-plus.org>

Unterstützung durch Hervorhebung der Syntax (Syntax-Highlighting) und automatisches Einrücken zur Strukturierung. Der Funktionsumfang dieser Editoren ist nicht nur auf HTML eingeschränkt, weswegen die Unterstützungen auch nicht HTML-spezifisch sind. Ein vollständiger „What you see is what you get“-Editor (WYSIWYG), der ohne Kenntnis von HTML genutzt werden kann, ist nicht im Sinne des Kerncurriculums.

Welche Unterstützungen sind für den Informatikunterricht mit HTML wünschenswert? Dazu ist ein Blick auf Schwierigkeiten von Schülerinnen und Schülern im Bereich von HTML hilfreich. Im bisherigen Unterricht des Autors zeigten sich unter anderem folgenden Probleme der Schülerinnen und Schüler (die nummerierte Aufzählung gibt keine Gewichtung vor, sondern dient dem späteren Referenzieren):

1. Korrekte Syntax von HTML-Elementen
2. Bezug zwischen HTML-Quelltext und entstehendem Dokument bleibt unklar
3. Arbeit in drei Fenstern zeitgleich (HTML, CSS und Browser)
4. Korrekte Verknüpfung von HTML- und CSS-Dateien
5. Identifizierung von HTML-Knoten, die durch CSS-Regeln formatiert werden
6. Zusammenhang zwischen Inhalt, Struktur und Aussehen

Des Weiteren gestalten sich gestufte Hilfen schwierig, da diese in einem erneuten (evtl. analogen) Format angeboten werden und damit ein weiterer Arbeitsbereich für die Schülerinnen und Schüler entsteht. Diese Vielzahl kann einige Lernende überfordern und sollte deshalb von einem HTML-Editor besser gelöst werden.

Aus dem ersten Punkt wird deutlich, dass ein Syntax-Highlighting eine mögliche Unterstützung darstellt. Dies unterstützt Lernende aber lediglich dann, wenn diese schon eine Ahnung davon haben, wie sie ein HTML-Element schreiben können.

Der zweite und dritte Punkt bedingen sich gegenseitig. Durch die Arbeit in vielen verschiedenen Fenstern oder Tabs sind die Auswirkungen von Änderungen an den Dokumenten nicht direkt sichtbar, sondern werden erst über Umwege sichtbar gemacht. Dies stört den Arbeitsfluss der Lernenden durch zusätzliche Routinen, die bei jeder Änderung ausgeführt werden müssen.

Punkt vier stellt gerade zu Beginn des Themenbereichs ein großes Problem dar, das viele Schülerinnen und Schüler vor eine zusätzliche Hürde stellt. Die Lernenden sollten am Ende der Unterrichtsreihe selbst eine solche Verknüpfung herstellen können, für die erste Begegnung mit Unterrichtsgegenstand ist dies jedoch ein vermeidbares Hindernis.

Die beiden letzten Punkte hängen ebenfalls zusammen, da die Identifizierung von HTML-Knoten, die durch eine CSS-Regel verändert werden, über die Struktur erfolgt, die den Lernenden jedoch nicht zu jeder Stelle klar ist. Zwar können sie einen Strukturbaum zu ihrem HTML-Dokument erstellen, nach einer Änderung am Dokument

muss dieser aber überarbeitet werden, so dass der Strukturbaum parallel weitergepflegt werden muss.

Im Zuge der Differenzierung wäre es ebenfalls hilfreich, wenn der HTML-Editor direkt die Aufgabenstellung und zusätzliche Hilfen anzeigen könnte, damit alle nötigen Arbeitsmaterialien an einem Platz sind.

### 3 Vorbilder

#### 3.1 Codefights

Auf der Online-Plattform [codefights.com](https://codefights.com) können Nutzer sich anmelden und verschiedene tägliche oder wöchentliche Herausforderungen lösen.

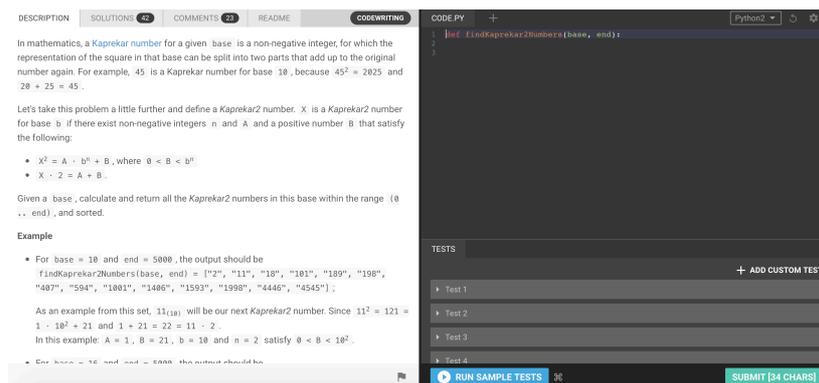


Abbildung 1: Beispielhafte Herausforderung bei Codefights

Dabei wird im Browser die Aufgabenstellung und ein Code-Editor angezeigt. Die Herausforderung lässt sich in einer Sprache der Wahl lösen und kompilieren. Glaubt man die Herausforderung gelöst zu haben, kann man Tests laufen lassen, deren Ergebnis angezeigt wird. Dieses HTML-fremde Beispiel zeigt einen Ansatz für Editoren innerhalb von Web-Browsern auf.

Dieser Ansatz bietet einige Vorteile gegenüber der einfachen Editorlösung. Eine Aufgabenstellung wird dauerhaft angezeigt, die Programmierung erfolgt direkt im gleichen Fenster. Es werden Tests angezeigt, mit denen Lösungen überprüft werden können. Zusätzlich bietet die Umsetzung in HTML und JavaScript einen Vorteil für die Plattformunabhängigkeit des Editors.

### 3.2 Mozilla Thimble

Verfolgt man diesen Ansatz weiter, wird man auf das Projekt Thimble von Mozilla Firefox aufmerksam, das einen ähnlichen Ansatz verfolgt und für HTML umsetzt.

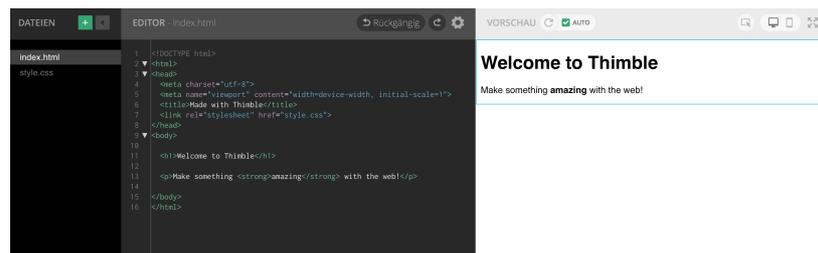


Abbildung 2: Mozilla Thimble

In einem Editor (Bereich in der Mitte) können HTML-Elemente eingegeben werden, die direkt in die Vorschau auf der rechten Seite übernommen werden. Man kann die Entstehung des Dokuments verfolgen und Auswirkungen von HTML-Elementen direkt beobachten. In der linken Liste kann zwischen HTML und CSS gewechselt werden. Auch innerhalb der CSS-Codes werden Auswirkungen unmittelbar in der Vorschau angezeigt.

Thimble wirkt einigen der angesprochenen Probleme entgegen. Dennoch ist der Einsatz in der Schule nicht ohne Weiteres umsetzbar, da für die Nutzung ein Nutzerkonto jeder Schülerin und jedes Schülers vorhanden sein muss. Zusätzlich liegen die Daten der Lernenden auf einem Cloud-Server außerhalb Deutschlands. Beide Argumente konnten datenschutzrechtlich für die Schule nicht restlos geklärt werden, so dass diese Möglichkeit nicht in Betracht gezogen wurde.

## 4 Eigenentwicklung

### 4.1 Überblick

Aus den beiden vorgestellten Editoren ergab sich eine grobe Richtung, in die eine Eigenentwicklung gehen sollte. Basierend auf HTML und JavaScript wurde ein eigener HTML-Editor geschrieben.

The screenshot shows the HTML-Editor interface with three tabs: 'Inhalt', 'Aussehen', and 'Struktur'. The 'Inhalt' tab is active, displaying the following HTML source code:

```

1 <html>
2 <head>
3 </head>
4 <body>
5 <h1>Hypertext Markup Language</h1>
6 <h4>Inhaltsverzeichnis</h4>
7 <ol>
8 <li>Entstehung</li>
9 <li>Syntax</li>
10 <ul>
11 <li>Elemente</li>
12 <li>Attribute</li>
13 </ul>
14 <li>HTML-Struktur</li>
15 </ol>

```

To the right, a preview window titled 'So würde ein Browser dein Dokument anzeigen' shows the rendered output:

**Hypertext Markup Language**

**Inhaltsverzeichnis**

- Entstehung
- Syntax
  - Elemente
  - Attribute
- HTML-Struktur

Vor der Entwicklung des World Wide Web und dessen Bestandteilen, zu denen auch HTML gehört, war es nicht möglich, Informationen auf digitalem Weg einfach, schnell und strukturiert zwischen mehreren Personen auszutauschen. Man benötigte neben Übertragungsprotokollen auch eine einfach zu verstehende Textauszeichnungssprache. Genau hier lag der Ansatzpunkt von HTML. Um Forschungsergebnisse mit anderen Mitarbeitern der Europäischen Organisation für Kernforschung (CERN) zu teilen und von den beiden Standorten in Frankreich und in der Schweiz aus zugänglich zu machen, entstand 1989 am CERN ein Projekt, welches sich mit der Lösung dieser Aufgabe beschäftigte. Am 3. November 1992 erschien die erste Version der HTML-Spezifikation.

Abbildung 3: HTML-Editor

Die Anleihen zum Thimble sind erkennbar. Auch dieser Editor bietet die Aufteilung in Quelltext und Vorschau und genau wie Thimble werden Änderungen im HTML-Quelltext direkt in der Vorschau umgesetzt. Der HTML-Editor weist ein rudimentäres Syntax-Highlighting auf und macht auf nicht geschlossene HTML-Tags aufmerksam. Eine Syntax-Vervollständigung wird nicht zur Verfügung gestellt.

Oberhalb des Editors werden drei Tabs angezeigt, mit denen Schülerinnen und Schüler zwischen HTML und CSS wechseln können. Änderungen im CSS-Editor werden ebenfalls live in der Vorschau angepasst. Die Verknüpfung von HTML und CSS wird implizit vom Programm vorgenommen.

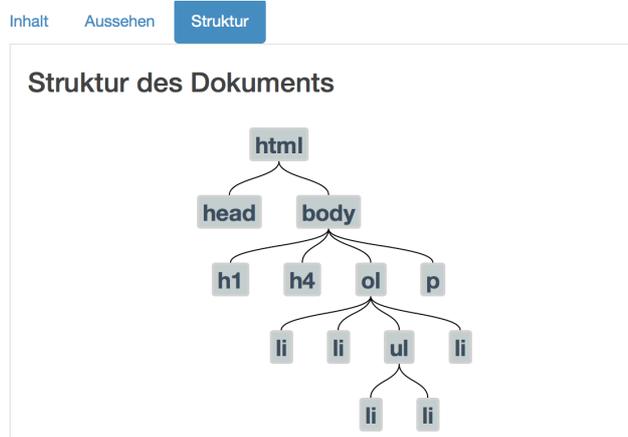


Abbildung 4: Strukturbaum des HTML-Dokuments

Der Editor bietet einen dritten Tab *Struktur*. Damit kann der aktuelle Strukturbaum des HTML-Dokuments angezeigt werden. Wird der HTML-Quelltext geändert, wird diese Änderung dynamisch am Strukturbaum vorgenommen.

## 4.2 Bereitstellen der Aufgabenstellung

Eine Aufgabenstellung kann den Schülerinnen und Schüler auf unterschiedliche Weise präsentiert werden. Die einfachste Variante ist das Einblenden oberhalb des Editors. So haben die Schülerinnen und Schüler den Editor, die Vorschau und die jeweilige Aufgabenstellung direkt auf einen Blick. Die verschiedenen Editoren können mit unterschiedlichen Quelltexten vorbelegt werden, um den Lernenden ein Template zum Arbeiten vorzugeben.

Die stärker leitende und damit für den Anfang besser geeignete Variante setzt auf kontextabhängige Aufgabenstellungen. Ein Beispiel: die Schülerinnen und Schüler erhalten zu Beginn ein leeres HTML-Dokument mit der übergreifenden Aufgabe, das HTML-Grundgerüst zu erstellen. Dazu erhalten sie den weiteren Hinweis einen öffnenden und einen schließenden Tag mit dem Namen `html` zu erstellen. Nachdem sie dies ausgeführt haben, wird der Schritt abgehakt und der nächste Schritt (Erstellung eines `head`-Elements) angezeigt. Somit wissen die Schülerinnen und Schüler jederzeit, was ihr nächster Schritt ist und können so zu einem vollständigen HTML-Grundgerüst gelangen.

Es ist ebenfalls ein Hybrid aus diesen beiden Varianten möglich, so dass die Schülerinnen und Schüler selbständig entscheiden können, welche Hilfen sie in Anspruch nehmen.

## 4.3 Bereitstellung von Hilfen

Um einem differenzierenden Unterricht gerecht zu werden, wurde der Editor erweitert, so dass gestufte Hilfen ermöglicht werden. Über verschiedene Buttons können die Schülerinnen und Schüler Hilfen zur Aufgabenstellung einblenden. Dies lässt sich ebenfalls an den Kontext anpassen, so dass den Schülerinnen und Schülern an verschiedenen Abschnitten ihrer Lösung auch verschiedene Hilfen angeboten werden.

Speziell für den Beginn der Behandlung von HTML ist das Grundgerüst zunächst eine Hürde. Deshalb lässt es sich ausschalten, so dass es für die Schülerinnen und Schüler unsichtbar ergänzt wird.

Ebenfalls für den Anfang kann eine Unterstützung bei der Erstellung von HTML-Elementen helfen. Dazu wurde ein Formular implementiert, das die Eigenschaften des gewünschten Elements abfragt und dann in den Quelltext einfügt.

Elementname

Attributname + weiteres Attribut

Attributswert

Abbildung 5: Formular zur Erstellung von HTML-Elementen

Damit lassen sich ebenfalls die Fachbegriffe der Bestandteile eines HTML-Elements festigen. Dieses Formular ist auch für CSS-Regeln verfügbar.

Um die Anwendung von CSS-Regeln zu visualisieren, können Schülerinnen und Schüler den Strukturbaum verwenden. Um die angewandten Regeln besser deutlich zu machen, werden Veränderungen am Attribut *color* auf Wunsch im Strukturbaum angezeigt. Dies fördert das Verständnis des Zusammenhangs zwischen den Bereichen Inhalt, Struktur und Aussehen.



Abbildung 6: Visualisierung der CSS-Regeln am Strukturbaum

## 5 Einsatz im Unterricht

Der vorgestellte HTML-Editor wurde zu Beginn des Themenbereichs HTML-Projekt eingesetzt. Um ein generelles Verständnis für die Beschreibungssprache zu fördern, erhielten die Schülerinnen und Schüler den HTML-Editor ohne die Tabs Aussehen und Struktur. Innerhalb des HTML-Editors befand sich reiner Text eines Wikipedia-Artikels. Das HTML-Grundgerüst wurde automatisch vom Editor ergänzt, so dass die Lernenden direkt mit dem Text arbeiten konnten.

In einem Unterrichtsgespräch wurden die Formatierungen gemeinsam erarbeitet und direkt im Text umgesetzt. So konnten die Schülerinnen und Schüler den direkten Zusammenhang zwischen dem ursprünglichen Text und der Erweiterung durch die HTML-Tags erkennen. Ferner wurden die Auswirkungen der HTML-Elemente den Lernenden direkt deutlich und sie konnten es von sich aus beschreiben.

Nach diesem Einstieg wurde den Schülerinnen und Schülern erläutert, dass in HTML-Dokumenten ein Grundgerüst nötig sei. Dies sollten die Schülerinnen und Schüler über die in 4.3 beschriebene, kontextabhängige Hilfe erarbeiten.

Im nächsten Schritt wurde die Struktur von HTML-Dokumenten behandelt. Die Lernenden erhielten einen neuen Editor, bei dem der Tab Struktur freigeschaltet war. Über den Vergleich von HTML-Dokument und erstelltem Dokumentenbaum war es den Schülerinnen und Schülern möglich, Regeln zur Erstellung eines Dokumentenbaums aufzustellen. Die Auswirkungen von Veränderungen am HTML-Dokument konnten direkt im Dokumentenbaum untersucht werden.

In einem nächsten Schritt wurde der Tab Aussehen (CSS) freigeschaltet. Über die Vorbelegung des CSS-Editors mit einer Regel konnten die Schülerinnen und Schüler die Folgen der CSS-Regel direkt im Dokument beobachten. Analog entwickelten sie eigene Regeln, die sich zunächst nur auf Tag-Namen bezogen. Der Bezug zwischen Dokumentenbaum und den CSS-Regeln konnte durch die eingeblendeten Hilfen hergestellt werden. Die Schülerinnen und Schüler konnten durch die Beobachtungen, die sich immer auf ihre eigenen Beispiele bezogen, selbst formulieren, wann eine CSS-Regel auf ein HTML-Element angewandt wird.

## Aufgabenstellung

Du hast ein neues Dokument erhalten. Wieder gibt es ein Layout-Dokument. Über den Menüpunkt "Struktur" kannst du dir den Strukturbaum einblenden.

Notiere dir auf einem Blatt:

1. Begründe anhand des Strukturbaums, warum das Dokument so formatiert wurde.
2. Schreibe einen Befehl, der alle ul-Listen grün färbt. Interpretiere das Ergebnis anhand des Strukturbaums.
3. Formuliere einen Merksatz, der die Regeln für das Anpassen von HTML-Elementen beschreibt. Du kannst dir Formulierungshilfen einblenden.
4. \*\*\*Füge der Formatierung für ol-Listen die Zeile "font-size:14pt;" hinzu und interpretiere das Ergebnis anhand des Strukturbaums.\*\*\*

Hilfe 1
Hilfe 2
Hilfe 3
Hilfe 4

Inhalt
Aussehen
Struktur

**Inhalt**

```

4      <body>
5      <h1>Hypertext Markup Language</h1>
6      <h4>Inhaltsverzeichnis</h4>
7      <ol>
8          <li>Entstehung</li>
9          <li>Syntax</li>
10         <ul>
11             <li>Elemente</li>
12             <li>Attribute</li>
13         </ul>
14         <li>HTML-Struktur</li>
15     </ol>
16     <p> Vor der Entwicklung des World Wide Web und desse
17 </body>
18 </html>
```

**So würde ein Browser dein Dokument anzeigen**

## Hypertext Markup Language

**Inhaltsverzeichnis**

1. Entstehung
2. Syntax
  - Elemente
  - Attribute
3. HTML-Struktur

Vor der Entwicklung des World Wide Web und dessen Bestandteilen, zu denen auch HTML gehört, war es nicht möglich, Informationen auf digitalem Weg einfach, schnell und strukturiert zwischen mehreren Personen auszutauschen. Man benötigte neben Übertragungsprotokollen auch eine einfach zu verstehende Textauszeichnungssprache. Genau hier lag der Ansatzpunkt von HTML. Um Forschungsergebnisse mit anderen Mitarbeitern der Europäischen Organisation für Kernforschung (CERN) zu teilen und von den beiden Standorten in Frankreich und in der Schweiz aus zugänglich zu machen, entstand 1989 am CERN ein Projekt, welches sich mit der Lösung dieser Aufgabe beschäftigte. Am 3. November 1992 erschien die erste Version der HTML-Spezifikation.

Abbildung 7: Komplette Lernumgebung

Nachdem die Grundlagen mit dem Editor gelegt wurden, sollten die Schülerinnen und Schüler in Gruppen eine eigene HTML-Seite erstellen. Dazu wurden die traditionellen

Editoren (siehe 2) zur Verfügung gestellt, die von einigen Lernenden genutzt wurden. Ungefähr die Hälfte der Lerngruppe verblieb aber in dem vorgestellten Editor und kopierte anschließend den Text aus den Dokumenten und fügten ihn im Windows-Editor ein.

Die Schülerinnen und Schüler hatten keine Schwierigkeiten mit dem Editor zu arbeiten und fanden sich in den verschiedenen Stufen gut zurecht. Es traten jedoch Probleme mit dem Internet-Explorer auf, die während der Durchführung nicht behoben werden konnten, so dass auf einen anderen Browser zurückgegriffen werden musste. Die kontextabhängige Hilfe erwies sich ebenfalls als fehleranfällig, was aber behoben werden konnte. So zeigte die Hilfe fälschlicherweise nicht den Abschluss einer Zwischenaufgabe an, wenn mehrere Tags des gleichen Namens vorhanden waren. Des Weiteren trat einmal ein Absturz des Browsers auf, bei dem der bisherige Fortschritt der Schülerinnen und Schüler verloren war. Über die Cache-Möglichkeiten von HTML5 wurde ein Zwischenspeicher eingerichtet, der regelmäßig die Zwischenergebnisse der Schülerinnen und Schüler speichert, so dass die bisherige Arbeit nach einem Absturz verfügbar bleibt.

## 6 Ausblick

Der vorgestellte Editor stellt eine gute Möglichkeit dar, einen Editor im Unterricht zu verwenden, der die Schülerinnen und Schüler durch geeignete Maßnahmen beim Erlernen von HTML und CSS unterstützt. Er gibt für den Einstieg verschiedene Darstellungsformen, um einen explorativen Ansatz bei der Arbeit mit HTML zu ermöglichen. In der Erprobung zeigten sich die Einsatzfähigkeit im Unterricht sowie kleinere Probleme, die behoben wurden. Schon während des Einsatzes wurden Erweiterungen umgesetzt, wie bspw. ein weiterer Tab, in dem JavaScript geschrieben werden kann.

Für die Zukunft sind noch Erweiterungen des Editors geplant. So soll es eine Export-Funktion geben, die aus den aktuellen Inhalten der Editoren HTML- und CSS-Dateien erstellt und zum Download anbietet. Die Visualisierung der CSS-Regeln wird im Dokumentenbaum nur beim Attribut *color* angezeigt. Bei der Weiterentwicklung sollen alle Regeln über ein Mouse-Over angezeigt werden. Dies ermöglicht eine noch intensivere Umsetzung mit dem Zusammenhang zwischen CSS und der Struktur des HTML-Dokuments. Des Weiteren soll die Erstellung von Aufgaben für Lehrende vereinfacht werden. Aktuell wird der Editor direkt in der JavaScript-Programmierung von Aufgabe zu Aufgabe angepasst, was sehr zeitintensiv ist.

Der Autor schreibt ein Konfigurations-Tool, mit dem Aufgaben, sichtbare Tabs, Hilfen usw. festgelegt werden können und daraus eine HTML-Datei generiert wird, die den Editor entsprechend umsetzt. Speziell die kontextabhängigen Hilfen haben sich in der Erprobung als sehr hilfreich erwiesen, beanspruchen in der Vorbereitung aber sehr viel Zeit. Mit dem einfacheren Konfigurations-Tool würde diese Hilfe einfacher umsetzbar.

Diese Erweiterung würde es ebenfalls ermöglichen, dass dieser Editor von anderen Lehrenden verwendet werden kann und auf deren jeweilige Unterrichtssituation angepasst werden kann. Momentan erfordert die Anpassung eine intensive Einarbeitung in die Programmierung des Editors.

### **Literaturverzeichnis**

- [Hes15] Hessisches Kultusministerium: Kerncurriculum gymnasiale Oberstufe – Gymnasium. Informatik (2015).