

Anleitung zur Erstellung Multimedialer Hilfekarten

Einsatz von Smartphones im mathematisch-naturwissenschaftlichen Unterricht

GABRIELE WOLFF – KAI WOLFF

Der Inhalt der Hilfekarten wird in HTML-Dateien beschrieben. HTML, die „Hypertext Markup Language“, ist eine textbasierte Auszeichnungssprache für digitale Dokumente wie zum Beispiel Webseiten. Die meisten Webseiten, die man im Internet findet, verwenden HTML. Wenn man eine Webseite im Browser aufruft (entweder durch Eingabe ihrer Adresse, der URL, oder durch Anklicken eines Links), liefert der Webserver eine HTML-Datei an den Browser. Die HTML-Datei enthält nicht die fertige Webseite, sondern den in HTML formulierten Quelltext, eine Art Bauplan. Der Browser interpretiert diesen Bauplan und setzt ihn je nach Umgebung um, so gut es geht.

Grundlagen des Webdesigns mit HTML, CSS und JavaScript

Webdesign ist ein schier unerschöpfliches Thema. Wir wollen es hier lediglich kurz anreißen und ein paar Grundlagen schaffen, deren Kenntnis die Erstellung der multimedialen Hilfekarten und den Einstieg in die weiterführende Literatur erleichtert. Unbedingt notwendig ist das jedoch nicht – wer lieber sofort mit den Hilfekarten beginnen möchte, überspringt dieses Kapitel und liest das „Kochrezept“.

```
<!-- Beispiel für eine einfache HTML-Datei -->
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Hilfekarte</title>
  </head>
  <body>
    <h1>HTML (Hypertext Markup
      Language)</h1>
    <p>HTML-Dokumente bestehen aus Text,
      in dem <strong>HTML-
      Elemente</strong> den Textteilen
      eine Struktur geben. Die meisten
      HTML-Elemente werden durch ein
      <strong>Tag-Paar</strong>
      markiert. <br>
      Im
      <a href="http://webkompetenz.wikidot.com/
        docs:html-handbuch">
        HTML5-Handbuch
      </a>
      von Stefan Münz findet man weitere
      Informationen.
    </p>
  </body>
</html>
```

Abb. 1. Beispiel für eine einfache HTML-Datei

HTML (Hypertext Markup Language)

HTML-Dokumente sind einfache Textdateien und haben Dateinamen, die normalerweise auf .HTML enden. Zur Bearbeitung genügt ein einfacher Texteditor, wie er in allen Betriebssystemen enthalten ist, komfortabler geht es jedoch mit einem Quelltext-Editor (unter Windows z. B. Notepad++, s. HO, 2015).

HTML-Dokumente bestehen aus Text, in dem **HTML-Elemente** den Textteilen eine Struktur geben. Die meisten HTML-Elemente werden durch ein **Tag-Paar** (engl. *tag*, Etikett) markiert. Ein Tag-Paar besteht aus einem Starttag `<...>` und einem Endtag `</...>`, die zusammen den Inhalt umschließen. Tag-Paare können auch geschachtelt werden. Die Einrückungen in Abbildung 1 verdeutlichen die Tag-Schachtelung dort, dienen aber nur der Übersichtlichkeit. Gleiches gilt für die verschiedenen Farben und Schriftstile. Sie erscheinen nur in der Darstellung des HTML-Dokuments durch den Quelltext-Editor, der diese Syntax-Hervorhebungen automatisch wählt. Auf die spätere Darstellung im Browser haben diese keinen Einfluss. Einrückungen und unterschiedliche Schriftstile in der Browserdarstellung des HTML-Dokuments erreicht man mit speziellen HTML-Elementen, die gleich erklärt werden. Abbildung 2 stellt die Struktur, die das HTML-Dokument durch die Schachtelung der Tags erhält, schematisch dar.

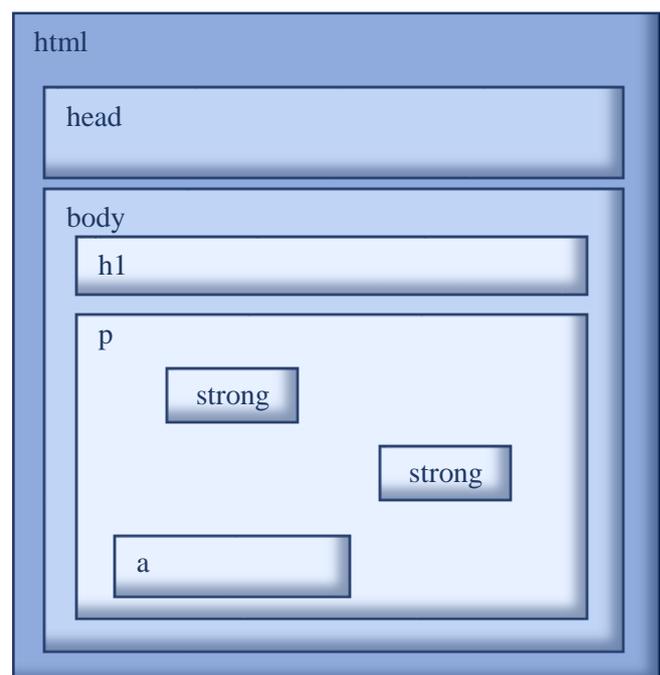


Abb. 2. Struktur des HTML-Dokuments: Schachtelung der Tags

Das Beispieldokument aus Abbildung 1 beginnt mit einer Kommentarzeile, die der Quelltext-Editor grün darstellt. Ein HTML-Kommentar beginnt immer mit `<!--` und endet mit `-->` und kann sich auch über mehrere Zeilen erstrecken. Ein Kommentar wird vom Browser ignoriert und der Inhalt nicht zur Anzeige gebracht. Trotzdem sind Kommentare sinnvoll, um z.B. Autor und Erstellungsdatum, Quellen und Erläuterungen unterzubringen. Da auch HTML-Elemente bis zum Kommentarende überlesen werden, kann man mit Kommentaren ganze Teile des HTML-Dokuments temporär auskommentieren und damit unwirksam machen. Auf diese Weise kann man etwa verschiedene Varianten ausprobieren.

Die Zeile `<!DOCTYPE html>` kennzeichnet die Datei intern als HTML-Dokument.¹ Das Tag-Paar `<html>...</html>` schließt dann den HTML-Inhalt ein, der aus zwei Teilen besteht. Im Dokumentkopf `<head>...</head>` wird der im Quelltext verwendete Zeichensatz angegeben und der Titel der Seite festgelegt, den der Browser in der Fensterüberschrift oder auf dem Reiter anzeigt (Abb. 3). Im Browserfenster wird der Rumpfteil `<body>...</body>` des HTML-Dokuments dargestellt.

Durch den Einschluss in `<h1>` und `</h1>` ist der Text dazwischen als Überschrift (erster Stufe) ausgezeichnet, und der Browser wählt dafür eine größere Schrift und Fettdruck (vgl. Abb. 3). Diese Hervorhebung ist jedoch nicht der Zweck des Überschrift-Tags, sondern eine Folge der Auszeichnung (engl. *markup*) als Überschrift. Dadurch erhält dieser Text eine Bedeutung, die über seinen Inhalt hinausgeht und zur semantischen Strukturierung des Textes beiträgt. Die Formatierung, also die konkrete visuelle Gestaltung des Textes, wird durch die HTML-Elemente nicht festgelegt und kann je nach Browser und Bildschirmgröße unterschiedlich ausfallen.

Der folgende Text zwischen `<p>` und `</p>` wird als Absatz (engl. *paragraph*) formatiert. Die Zeilenumbrüche dieses Textes sind (wie auch schon bei der Überschrift) nur von der Breite des Browserfensters abhängig und können nicht durch Umbrüche im Quelltext festgelegt werden. Mit dem Tag `
` kann man aber einen Zeilenumbruch (engl. *break*) an einer festen Stelle erzwingen (beim `
`-Tag kann man auf das Endtag verzichten). Der in `...` eingeschlossene Text wird dadurch als besonders wichtig gekennzeichnet und – als Nebeneffekt – in der Browserdarstellung fettgedruckt.

Mit dem `<a>`-Tag definiert man einen **Hyperlink**, einen Querverweis, der zu einer anderen Stelle im aktuellen Dokument oder zu einem anderen Dokument im Internet führt. Über solche Links sind die Dokumente im Internet verbunden – alle Links zusammen weben das „World Wide Web“. Den eingeschlossenen Text (im abgebildeten Beispiel „HTML5-Handbuch“) stellt der Browser farbig und unterstrichen dar und kennzeichnet ihn so als anklickbar (vgl. Abb. 3). Ein Klick darauf, und der Browser zeigt das verlinkte HTML-Dokument an. Über den Zurück-Button des Browsers kommt man wieder zum Ausgangsdokument zurück. Die URL der Linkadresse für einen Hyperlink gibt man im `href`-Attribut des Starttags an.

Die Entwicklung der Hypertext Markup Language (HTML) ist eng mit der Entwicklung des Internets verbunden. Die erste Version des Sprachstandards stammt aus 1992 und geht auf eine Initiative des CERN zurück mit dem Ziel, digitale Dokumente

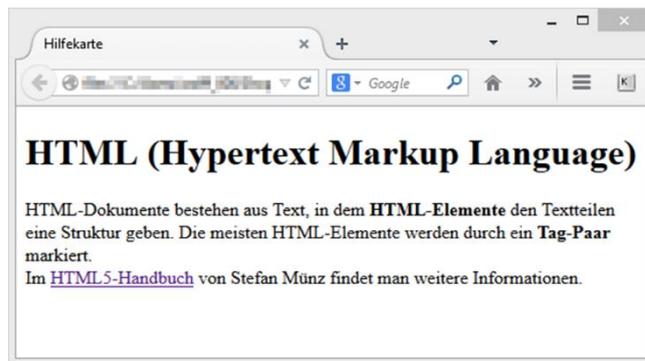


Abb. 3. Darstellung der HTML-Datei aus Abbildung 1 im Browser

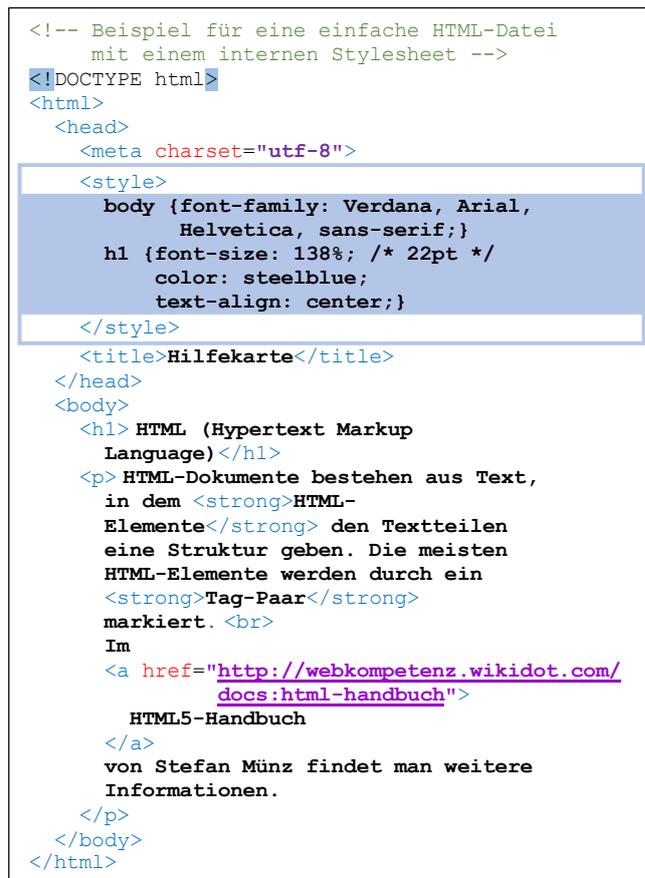


Abb. 4. HTML-Datei mit internem CSS-Stylesheet im `<head>`-Bereich

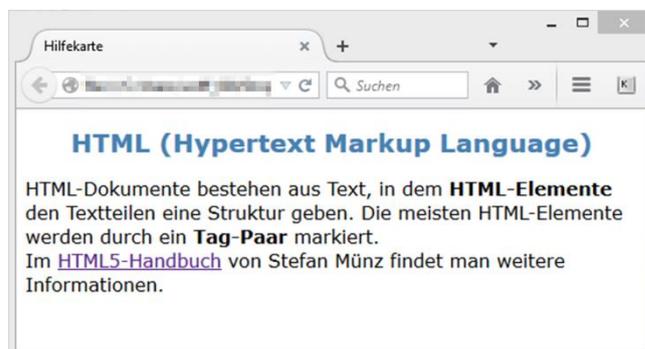


Abb. 5. Browserdarstellung der HTML-Datei aus Abbildung 4

¹ Anhand der Dateinamen-Erweiterungen `.html` wird nur gesteuert, dass die betreffende Datei nach einem Doppelklick zum Browser geschickt wird. Erst die `<!DOCTYPE html>`-Zeile veranlasst den Browser, den Dateiinhalt als HTML-Dokument zu interpretieren.

```

/* style.css */
body {font-family: Verdana, Arial, Helvetica,
      sans-serif;}
h1 {font-size: 138%; /* Schriftgröße 22pt */
    color: steelblue;
    text-align: center;}

```

Abb. 6. Inhalt des externen Stylesheets `style.css`

```

<!-- Beispiel für eine HTML-Datei
      mit externem Stylesheet -->
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <link href="style.css" rel="stylesheet">
    <title>Hilfeskarte</title>
  </head>
  <body>
    <h1>HTML (Hypertext Markup
      Language)</h1>
    <p>HTML-Dokumente bestehen aus Text,
      in dem <strong>HTML-
      Elemente</strong> den Textteilen
      eine Struktur geben. Die meisten
      HTML-Elemente werden durch ein
      <strong>Tag-Paar</strong>
      markiert. <br>
      Im
      <a href="http://webkompetenz.wikidot.com/
        docs:html-handbuch">
        HTML5-Handbuch
      </a>
      von Stefan Münz findet man weitere
      Informationen.
    </p>
  </body>
</html>

```

Abb. 7. HTML-Datei mit einem verlinkten, externen CSS-Stylesheet

```

<!-- Beispiel für eine HTML-Datei
      mit CSS-Regeln im style-Attribut -->
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8">
    <title>Hilfeskarte</title>
  </head>
  <body style="font-family: Verdana, Arial,
    Helvetica, sans-serif;">
    <h1 style="font-size: 138%;
    color: steelblue;
    text-align: center;">
    HTML (Hypertext Markup
      Language)</h1>
    <p>HTML-Dokumente bestehen aus Text,
      in dem <strong>HTML-
      Elemente</strong> den Textteilen
      eine Struktur geben. Die meisten
      HTML-Elemente werden durch ein
      <strong>Tag-Paar</strong>
      markiert. <br>
      Im
      <a href="http://webkompetenz.wikidot.com/
        docs:html-handbuch">
        HTML5-Handbuch
      </a>
      von Stefan Münz findet man weitere
      Informationen.
    </p>
  </body>
</html>

```

Abb. 8. HTML-Datei mit CSS-Regeln im `style`-Attribut

mithilfe einer Auszeichnungssprache zu strukturieren. Für die multimedialen Hilfekarten verwenden wir die aktuelle Version HTML5 (WORLD WIDE WEB CONSORTIUM (W3C), 2014). Gute Einführungen und Referenzen findet man im bereits erwähnten HTML5-Handbuch (MÜNZ, 2014) und im HTML5-Tutorial der W3Schools (REFSNES DATA, o. J.). Gedruckte Bücher haben in diesem dynamischen Umfeld den Nachteil, dass sie sehr schnell veralten. Es gibt aber eine Reihe gute deutsche Bücher, die aktuell sind und als Lehr- und Nachschlagewerke dienen können (z. B. WOLF, 2016).

CSS (Cascading Stylesheets)

Die Standarddarstellung durch den Browser, wie sie in Abbildung 3 gezeigt wird, kann durch CSS verändert werden. Dazu kann man innerhalb des `<head>`-Bereiches und eingeschlossen in das Tag-Paar `<style>...</style>` eigene CSS-Regeln einfügen. In Abbildung 4 ist ein kleines **CSS-Stylesheet** mit zwei Regeln hellblau unterlegt.

Jede CSS-Regel beginnt mit einem **Selektor**, mit dem man das HTML-Element angibt, dessen Gestaltung beeinflusst werden soll. Danach folgen, in geschweifte Klammern eingeschlossen, eine oder mehrere **Deklarationen**. Jede Deklaration besteht aus einem Eigenschaft-Werte-Paar und wird mit einem Semikolon beendet.

In der ersten CSS-Regel im Beispiel aus Abbildung 4 ist der Selektor `body`, die nachfolgende Deklaration verändert also die Darstellung des gesamten HTML-Rumpfes zwischen `<body>` und `</body>`. Mit der Eigenschaft `font-family` wird die Schriftart gesteuert. In der Standarddarstellung wählt der Browser eine Serifenschrift, stattdessen soll er nun eine serifenlose Schrift verwenden. Da wir nicht wissen, auf welchem Ausgabegerät und mit welchem Browser die HTML-Datei dargestellt werden wird, geben wir als Eigenschaftswert eine Liste von Schriftnamen an; der Browser wählt dann die erste verfügbare.

Die zweite CSS-Regel bezieht sich auf die Darstellung der Überschriften (erster Stufe). Wir verändern die Schriftgröße auf 138 % der Grundschriftgröße, wählen als Farbe Stahlblau und setzen Überschriften zentriert. Ausgehend von einer Grundschriftgröße von 16 pt wird der Browser die Überschriften also mit 22 pt (= 16 pt · 138 %) darstellen. Dieser Hinweis steht auch in einem Kommentar; `/*` und `*/` markieren CSS-Kommentare.

Das Ergebnis sieht man in Abbildung 5. Der gesamte Text verwendet eine serifenlose Schrift. Die Überschrift steht zentriert und in Blau und ist 22 pt groß (aber das könnte man nur im Vergleich beurteilen). Auch die Überschrift ist serifenlos, obwohl es keine entsprechende Deklaration in der CSS-Regel für `h1` gab. Da die HTML-Elemente aber geschachtelt sind, wirkt die CSS-Regel für `body` auch auf alle HTML-Elemente, die in `<body>...</body>` eingeschlossen sind, solange es für diese keine eigene CSS-Regel gibt, die etwas anderes vorgibt.

Die CSS-Regeln (ohne das umgebende HTML-Tag-Paar `<style>...</style>`) kann man auch in ein **externes CSS-Stylesheet** auslagern. Dabei handelt es sich um eine Textdatei mit frei wählbarem Dateinamen und der Namensendung `.css`, wie sie Abbildung 6 zeigt. In der HTML-Datei entfällt dann der `<style>`-Block und wird dort durch ein `<link>`-Element ersetzt. Den Dateinamen der CSS-Datei (im Beispiel `style.css`) gibt man im `href`-Attribut an und kennzeichnet

die verlinkte Datei über das `rel`-Attribut als Stylesheet (vgl. Abb. 7). Durch die Verwendung eines externen CSS-Stylesheets ändert sich nichts an der Browserdarstellung, die für unsere Beispieldateien wieder Abbildung 5 entspricht.

Als Linkadresse im `href`-Attribut haben wir nur den Dateinamen der Stylesheet-Datei selbst angegeben. Deshalb müssen HTML- und CSS-Datei im gleichen Verzeichnis liegen. Bei einem größeren Projekt mit mehreren CSS-Stylesheets sollte man schon aus Gründen der Übersichtlichkeit dort ein Unterverzeichnis `css` anlegen und die Stylesheets darin sammeln. Im `<link>`-Element müsste man entsprechend `href="css/style.css"` angeben.

Ein externes CSS-Stylesheet bietet auch den Vorteil, dass es – wie bei unseren Hilfekarten – in mehreren HTML-Dateien wiederverwendet werden kann. Das gewährleistet die Einheitlichkeit der Formatierung und reduziert außerdem die Größe der HTML-Dateien.

Als dritte Möglichkeit kann man CSS-Regeln auch direkt in das Start-Tag eines HTML-Elements schreiben und sie als Wert dessen `Style`-Attribut zuweisen (Abb. 8). Diese Methode wird manchmal auch **Inline-Style** genannt. Die Wirkung der CSS-Regel ist dann auf das betreffende HTML-Element beschränkt.

Alle drei Methoden, CSS-Regeln zu definieren – als externes CSS-Stylesheet, als internes Stylesheet im HTML-Head oder als Inline-Style – können miteinander kombiniert werden. Bei konkurrierenden Regeln „gewinnt“ diejenige, die als letzte vor dem zu gestaltenden Text steht:

- Inline-Styles haben höchste Priorität.
- Bei CSS-Regeln im HTML-Head entscheidet die Reihenfolge im Quelltext. Weiter unten stehende Regeln „überschreiben“ frühere Regeln – unabhängig davon, ob sie aus einem externen oder internen Stylesheet stammen.

Es empfiehlt sich, möglichst viele CSS-Regeln in externe Stylesheets zu schreiben. Inline-Styles sollten (außer zu Testzwecken) möglichst nicht verwendet werden.

CSS entstand als Ergänzung zu HTML ebenfalls am CERN. Ein erster Entwurf wurde 1996 veröffentlicht, die aktuelle Version, die beständig weiterentwickelt wird, ist CSS3 (WORLD WIDE WEB CONSORTIUM (W3C), 2011). In den W3Schools findet man ein gutes CSS-Tutorial (REFSNES DATA, o. J.), das HTML5-Handbuch enthält eine kurze Übersicht (MÜNZ, 2010). Eine sehr umfangreiche und fundierte Einführung in die Gestaltung von Webseiten mit HTML5 und CSS3 geben die Bücher „HTML5 und CSS3“ (WOLF, 2016) und „Einstieg in CSS“ (MÜLLER, 2014).

JavaScript

JavaScript ist eine Programmiersprache, die ursprünglich entwickelt wurde, um die eingeschränkten Möglichkeiten von HTML und CSS zu erweitern. Mit den aktuellen Webstandards HTML5 und CSS3 könnte man *Responsive Webdesign* (dazu gleich mehr) auch ohne JavaScript realisieren. Die Nutzung von Multimedia-Elementen in den Hilfekarten wird durch JavaScript jedoch sehr erleichtert, da sich die verschiedenen Browser auf den unterschiedlichen Geräten in diesem Bereich stark unterscheiden. Das bereits erwähnte Buch „HTML5 und CSS3“ enthält eine kurze Einführung in JavaScript und gibt Hinweise zur

Verwendung (WOLF, 2016, Kap. 17 ff.). Wenn man tiefer einsteigen möchte, kann man z. B. zu „Professionell entwickeln mit JavaScript“ (ACKERMANN, 2015) greifen.

Responsive Webdesign

Ein Ziel bei der Entwicklung der multimedialen Hilfekarten war ein Design, das sich automatisch an die Eigenschaften und Erfordernisse der verschiedenen Endgeräte anpasst. So unterschiedlich die Geräte in ihrer Bildschirmgröße, -auflösung und -orientierung und hinsichtlich der verfügbaren Eingabemethoden wie Maus oder Touchscreen auch sind – der grafische Aufbau und die Bedienung sollte dem *selbstständig* Rechnung tragen. Für dieses Ideal hat sich die Bezeichnung **Responsive Webdesign** eingebürgert (MARCOTTE, 2010). Besonders auffällig ist Responsive Design beim Layout der Seiten, das sich abhängig von der Größe und vom Format des Browserfensters verändert. Weniger offensichtlich sind die Anpassungen zum Beispiel bei den Bedienelementen für multimediale Inhalte, die auf allen Geräten ohne spezielle Vorkehrungen einfach funktionieren. Damit genügt eine einzige Version jeder Hilfekarte für alle Geräte.

Technische Basis dafür sind meist (wie auch bei unseren Hilfekarten) HTML5 und CSS3 in Kombination mit JavaScript. Zusammen bilden sie ein **Framework**, einen Rahmen, der hier ohne Detailkenntnisse genutzt werden kann. Aus einem vorgefertigten **Template**, einer Vorlage für eine HTML-Datei, entsteht durch inhaltliche Anpassungen ein eigener Satz von Hilfekarten. Die CSS- und JavaScript-Anteile des Frameworks müssen dort lediglich eingebunden werden (das ist im Template bereits vorbereitet) und wirken im Hintergrund. Ein empfehlenswertes Buch zum Thema ist „Responsive Webdesign – Anpassungsfähige Websites programmieren und gestalten“ (ERTEL & LABORENZ, 2015)

Einrichten der Hilfekarten – ein Kochrezept

Wenn Sie diese Anleitung lesen, haben Sie das Materialpaket bereits heruntergeladen und entpackt (vgl. Abb. 9). Öffnen Sie bitte den darin enthaltenen Dateordner `beispiel_genetik` und vergleichen den Inhalt mit Abbildung 10. Hier finden Sie Beispiel-Hilfekarten zu einer Humangenetik-Unterrichtsreihe, die Sie sofort ausprobieren können. Öffnen Sie eine der HTML-Dateien mit einem Doppelklick im Browser:

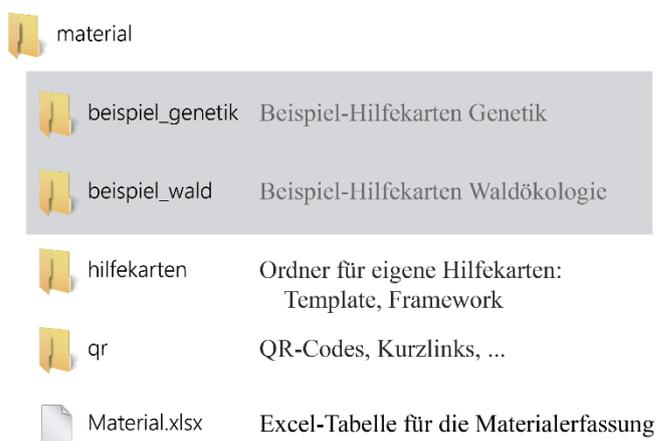


Abb. 9. Inhalt des `material`-Ordners aus dem Materialpaket

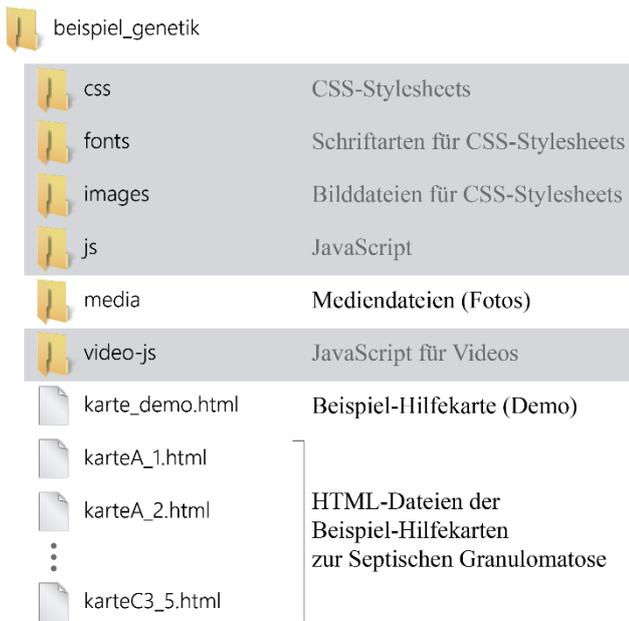


Abb. 10. Inhalt des *beispiel_genetik*-Ordners aus dem Materialpaket mit funktionsfähigen Beispiel-Hilfekarten zum Ausprobieren

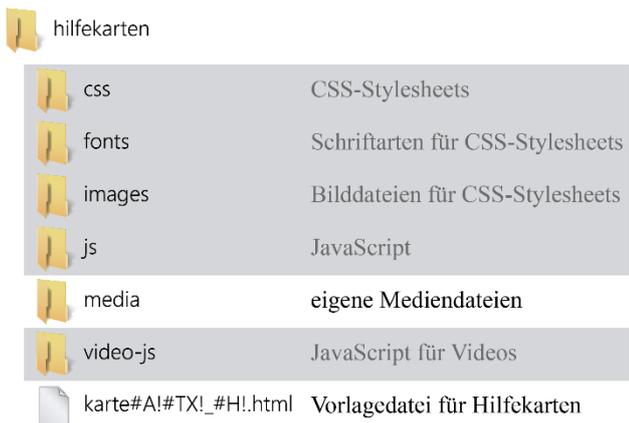


Abb. 11. Inhalt des *hilfekarten*-Ordners aus dem Materialpaket mit allen benötigten Framework-Ordnern (grau), einem vorbereiteten Ordner für eigene Mediendateien und einer Vorlage für die HTML-Hilfekarten

- In der Datei *karte_demo.html* sind viele Möglichkeiten der multimedialen Hilfekarten zu Demonstrationszwecken vereinigt. Sie finden dort zum Beispiel verschiedene Schriftschnitte und farbige Textauszeichnungen, mehrere Videos und Tonaufnahmen. Diese Karte ist als technische Referenz gedacht, nicht als Vorlage für eine echte Hilfekarte, dafür ist sie viel zu überladen.
- Die Dateien *karteA1_1.html* bis *karteC3_5.html* sind Hilfekarten für das Unterrichtsprojekt zur Septischen Granulomatose.
- Der Ordner *media* enthält Bilddateien, die in den Beispiel-Hilfekarten verwendet werden.
- Die in der Abbildung grau unterlegten Ordner gehören zum Framework und müssen nicht weiter beachtet werden. Wenn Sie die Hilfekarten-Dateien im Dateisystem verschieben oder die fertigen Hilfekarten später auf einen Webserver übertragen, kopieren Sie die Framework-Ordner immer unverändert und vollständig mit. Sie sind für die Funktionsfähigkeit der Hilfekarten unbedingt erforderlich.

Das Materialpaket enthält einen weiteren Ordner namens *beispiel_wald* mit Hilfekarten zu einer weiteren Unterrichtsreihe. Im Ordner *hilfekarten* finden Sie alle Zutaten für die Zubereitung individueller Hilfekarten. In diesem Ordner werden nachher Ihre eigenen Hilfekarten-Dateien entstehen. Öffnen Sie bitte den Ordner und vergleichen ihn mit Abbildung 10.

- Die Datei *karte#A!#TX!_#H!.html* ist eine Vorlage, aus der Sie individuelle Hilfekarten für Ihr Unterrichtsprojekt erstellen können. Wie das geht, wird gleich beschrieben.
- Der Ordner *media* ist zur Aufnahme eigener Mediendateien gedacht und momentan noch leer. Wenn Sie z.B. für Ihre Hilfekarten Grafiken erstellen, können Sie die Bilddateien hier ablegen.
- Die übrigen, in der Abbildung grau unterlegten Ordner gehören zum Framework. Sie müssen immer in einem gemeinsamen Ordner mit den HTML-Dateien der Hilfekarten liegen und dürfen auch nicht umbenannt werden.

Für jede Hilfekarte muss eine eigene HTML-Datei erzeugt werden. Ausgangspunkt ist die Vorlagedatei, die Platzhalter für die individuellen Inhalte enthält. Dann trägt man diese Inhalte (Titel, Texte, Bilder, Videos etc.) zusammen und füllt damit die Vorlage aus. Im Materialpaket finden Sie dazu eine Tabellenvorlage (*Material.xlsx*).

1. Tragen Sie bitte das Material für jede geplante Hilfekarte zusammen und füllen die Materialtabelle aus. Tabelle 1 enthält als Beispiel die Daten für die Hilfekarte aus Abbildung 12. Mit den dort erfassten Daten wird dann die mitgelieferte HTML-Vorlage angepasst. Dazu wird jede Textvariable durch den entsprechenden Inhalt ersetzt.
2. Kopieren sie die Datei *karte#A!#TX!_#H!.html* aus dem Materialpaket und geben ihr den neuen Namen *karteC2_5.html*, indem sie die Textvariable *#A!* durch



Abb. 12. Multimediale Hilfekarte mit einem Video auf einem Smartphone

- C ersetzen, #TX! durch 2 und #H! durch 5 – so wird die jede Hilfskarte eindeutig einer HTML-Datei zugeordnet.
3. Öffnen Sie danach die neue Hilfskartendatei in einem Texteditor und passen den Inhalt an. Die Anpassung selbst besteht aus einfachen Textersetzungen: Jede Textvariable #...! im Quelltext der Vorlage wird durch den entsprechenden Inhalt aus der Materialtabelle ersetzt. Mit der Suchen/Ersetzen-Funktion des Editors sucht man also nacheinander alle Textvariablen aus Tabelle 1 (#UE_TITEL!, #UE_RECHT!, etc.) und ersetzt sie durch die jeweiligen Inhalte in der mittleren Tabellenspalte.
 4. Eine Hilfskarte soll immer einen Text enthalten, optional sind jedoch die Medienelemente Bild, Video und Audio. Wenn solche optionalen Elemente in einer Hilfskarte nicht verwendet werden, können ganze Abschnitte des Quelltextes gelöscht werden. Benötigt man mehrere gleichartige Medienelemente, so kopiert man die entsprechenden Abschnitte. Auch ist die Reihenfolge Bild – Video – Audio nicht starr und kann beliebig geändert werden.
 5. Die Hilfskarte aus Abbildung 12, die wir in Tabelle 1 als Beispiel genommen haben, enthält neben dem Text nur ein Video und weder Bild noch Audio. In der Hilfskartendatei können deshalb die Bild- und Audio-Abschnitte gelöscht werden, die man über die Suche nach #PI_OPT! und #AU_OPT! findet. Der Quelltext der Musterkarte enthält entsprechende Kommentare, die in <!-- und --> eingeschlossen sind.
Verwendet man ein kein Video, so können zwei Video-Abschnitte im Quelltext gelöscht werden. Man findet sie über #VI_OPT!.
 6. Wird ein Video eingesetzt, so sind genau in diesen #VI_OPT!-Abschnitten Anpassungen nötig. Im oberen dieser Abschnitte muss #VI_H! durch die Höhe und #VI_B! durch die Breite des Videos ersetzt werden. Dabei kommt es nur auf das Seitenverhältnis an, sodass man z.B. die Pixelmaße einsetzen kann. Stimmt das Format nicht exakt, so zeigen sich später außen schwarze Balken, Schlimmeres passiert nicht.
 7. Verwenden Sie ein eigenes Video, so kopieren Sie die Videodatei in den Ordner media. Die Textvariable #VI_URL! erhält dann die relative Pfadangabe zu dieser Videodatei, also etwa media/beispielvideo.mp4. Bei einem externen Video aus einer Internetquelle (Rechte beachten!) müssen Sie als URL dessen „Internetadresse“ http://... angeben (vgl. Tab. 1). Häufig finden Sie diese auf der Internetseite als Link-Adresse angegeben.
 8. Die Einbindung von Videos von Streaming-Plattformen wie YouTube und Dailymotion unterscheidet sich von anderen Videos. Die verschiedenen Varianten sind in den #VI_OPT!-Abschnitten vorbereitet. Die Stellen, an denen Anpassungen erforderlich sind, findet man dort über #VI_QUELLE. Im oberen Abschnitt wird für ein Video von einer Streaming-Plattform eine zusätzliche JavaScript-Bibliothek benötigt (s. HEFFERNAN, 2015). Man wählt die passende Zeile aus und löscht die übrigen – für Videos, die nicht von einer Streaming-Plattform kommen, wird keine weitere Bibliothek gebraucht.
Im unteren Video-Abschnitt gibt es für die verschiedenen Video-Typen unterschiedliche <video>...</video>-Blöcke, von denen man den passenden wählt und die anderen aus dem Quelltext löscht. Als URL für YouTube-Videos gibt man die Adresse an, die der Browser beim Betrachten der YouTube-Seite im Adressfeld angibt (für die anderen Streaming-Videos gilt Entsprechendes).
 9. Falls Ihre Hilfskarten Bild oder Audio verwendet, suchen Sie die entsprechenden Abschnitte (#PI_OPT! oder #AU_OPT!) und ersetzen die #Pi- oder #AU-Textvariablen. Die nicht benötigten Abschnitte sollten gelöscht werden. Eigene Bilder und Audios werden in den media-Ordner kopiert, als URL gibt man die relativen Pfade zu den Dateien an (z.B. media/bild.jpg oder media/musik.mp3).
 10. Die Einrichtung der Buttons, deren Anklicken zu den anderen Hilfskarten führt, wird durch eine JavaScript-Funktion sehr erleichtert. Es müssen im Aufruf der Funktion myButtons lediglich die Werte für die Textvariablen #A!, #T!, und #H! sowie #MAX! eingetragen werden.
 11. Sammeln Sie alle Ihre Hilfskarten im Ordner hilfeskarten zusammen mit den Ordnern, die zum Framework gehören (vgl. Abb. 11). Der Name dieses gemeinsamen Ordners selbst ist dabei nicht von Belang, wichtig ist nur, dass die HTML-Dateien der Hilfskarten und die Framework-Ordner in einem gemeinsamen Ordner liegen. Diese Struktur muss auch bei der Bereitstellung der Hilfskarten, etwa auf einem Webserver, beibehalten werden.
 12. Sobald eine Hilfskartendatei fertiggestellt ist, können Sie sie mit einem Doppelklick aufrufen und im Browser darstellen lassen. Prüfen Sie für alle Hilfskarten, ob Texte, Bilder, Videos und Audios erscheinen und richtig dargestellt und abgespielt werden. Wenn einzelne Medienelemente fehlen, kontrollieren Sie zunächst die angegebene URL auf Fehler. Pfade und Dateinamen müssen exakt stimmen, sonst „findet“ der Browser die Dateien nicht. Auf manchen Servern bereiten Dateinamen mit Sonderzeichen wie etwa Umlauten Probleme. Verwenden Sie daher für alle Dateinamen der Hilfskarten und Mediendateien nur die Zeichen a..z, A..Z, 0..9, Binde- und Unterstrich sowie einen einzigen Punkt vor der Namenserweiterung. Andere Fehlerquellen sind:
 - unpaarige Anführungszeichen. Die Werte, die HTML-Attributen zugewiesen werden, müssen in doppelte Anführungszeichen eingeschlossen sein. Fehlen die Anführungszeichen am Ende, so wird alles bis zu den nächsten Anführungszeichen zum Wert hinzugenommen – mit unerwünschten Folgen.
 - Fehler in HTML-Tags: Die Tags müssen immer in < und > eingeschlossen sein. Die meisten HTML-Elemente haben ein Start- und ein Ende-Tag; das Ende-Tag hat immer die Form </...>. HTML-Elemente können auch geschachtelt werden, dürfen sich aber nicht überschneiden – etwas wie <h3>Überschrift <p>Text</h3></p> ist nicht zulässig.
 - Flash-Videos (Video-Typ „flv“) bereiten gelegentlich Probleme, wenn sie aus lokal gespeicherten HTML-Dateien heraus aufgerufen werden und lassen sich dann nicht abspielen. Nach der Übertragung auf den Web-Server (s. nächstes Kapitel) funktionieren sie in der Regel.
 13. Die Buttons zum Aufruf anderer Hilfskarten funktionieren, sobald deren HTML-Dateien erstellt sind. Testen Sie, ob alle Verkettungen der Hilfskarten über die Buttons stimmen.

Bereitstellung der Hilfekarten

Der Einsatz der Hilfekarten im Unterricht setzt voraus, dass alle benötigten HTML-, Medien- und Framework-Dateien für die Webbrowser der Smartphones oder Laptops der Schülerinnen und Schüler „erreichbar“ sind. Dafür sind verschiedene Szenarien denkbar:

- Die Schule verfügt über eine WLAN-Infrastruktur, zu der die Schülerinnen und Schüler Zugang haben. Dann können die Hilfekarten im Dateisystem des Schulnetzes untergebracht werden, wobei der Netzadministrator sicher hilft.
- Viele Schulen haben eine eigene Website. Bitten Sie den Webmaster, die Hilfekartendateien in einen (auf der Website nicht verlinkten) Bereich zu kopieren. Dann sind die Hilfekarten auch über das Internet zu erreichen, wenn die URLs bekannt sind (dazu gleich mehr). Viele Schülerinnen und Schüler verfügen bereits über eine mobile Daten-Flatrate – kostengünstiger wäre natürlich ein Zugang über ein schuleigenes WLAN.
- Besteht nicht die Möglichkeit, die Hilfekarten auf der Schul-Website unterzubringen, benötigt man einen eigenen Webpace. Im Internet findet man eine Reihe von seriösen Anbietern von kostenlosem, werbefreiem Webpace. Da die Hilfekarten nur geringe technische Anforderungen stellen und auch kein hohes Datenaufkommen verursachen, sind die meisten dieser Angebote grundsätzlich geeignet. Mit den Zugangsdaten zum Webserver, die man nach der Anmeldung erhält, können die Hilfekartendateien mittels eines FTP-Programms (z. B. dem für viele Betriebssysteme verfügbaren FileZilla, s. KOSSE, o. J.) hochgeladen werden.

Nach der Übertragung sämtlicher Dateien sollte man sich noch einmal von der Funktionsfähigkeit der Karten in der Zielumgebung überzeugen. Die jeweils ersten Hilfekarten zu einer Aufgabe sollen die Schülerinnen und Schüler später mit den Angaben auf einer gedruckten QR-Karte aufrufen können.

Dazu benötigt man die URLs der Hilfekarten, die man nach dem Aufruf der Hilfekarte aus dem Adressfeld des Browsers kopiert und in einer Tabelle sammelt. Eine URL kann man sich als eine eindeutige Adresse der betreffenden HTML-Datei im Internet vorstellen. Wenn man eine URL im Adressfeld des Browsers eingibt oder einen Link mit dieser URL anklickt, so wird schließlich der Inhalt dieser HTML-Datei im Browser dargestellt. Wenn die HTML-Datei umbenannt oder in einen anderen Ordner verschoben wird, verliert die URL ihr Ziel und die Hilfekarte ist nicht mehr aufrufbar.

- Die Original-URL ist wahrscheinlich lang und unpraktisch einzutippen. Kostenlose Internetdienste erzeugen daraus Kurzlinks (z. B. GNU YORK IT DIENSTLEISTUNGEN, o. J.). Ruft man im Browser den Kurzlinks auf, so wird man von dort sofort zur ursprünglichen (langen) Linkadresse weitergeleitet. Einige Internetdienste können zusammen mit den Kurzlinks auch Deaktivierungslinks erzeugen. Nach einem Aufruf eines solchen Deaktivierungslinks funktioniert der Kurzlink nicht mehr.
- Alternativ kann man auf jeder QR-Karte einen QR-Code abdrucken. Mit einem QR-Code-Scanner auf dem Smartphone kann man diesen Code fotografieren und die darin enthaltene URL an den Browser schicken lassen. Das erspart die manuelle Eingabe der URL im Browser. Im Internet findet man kostenlose Dienste, die eine URL in einen QR-Code umwandeln und als Bilddatei zum Download

anbieten (z. B. ARSAVA, o. J.). Diese Bilddateien sammelt man in einem Dateiodner und gibt ihnen beim Abspeichern einen aussagekräftigen Namen, um die Zuordnung zu den Hilfekarten zu erleichtern.

Literatur

ACKERMANN, P. (2015). *Professionell entwickeln mit JavaScript: Design, Patterns, Praxistipps* (Rheinwerk Computing) (1. Auflage.). Bonn: Rheinwerk-Verlag (vormals Galileo Press).

ARSAVA. (o. J.). QR Code Generator | goqr.me. <http://goqr.me/de/#t=url> [18.2.2015].

ERTEL, A. & LABORENZ, K. (2015). *Responsive Webdesign – Anpassungsfähige Websites programmieren und gestalten*. Bonn: Galileo Press.

GNU YORK IT DIENSTLEISTUNGEN. (o. J.). Kurz URLs mit No Referer Funktion & Malware Schutz | t1p.de. <http://t1p.de/> [18.2.2015].

HEFFERNAN, S. (2015). Plugins · videojs/video.js Wiki · GitHub. <https://github.com/videojs/video.js/wiki/Plugins> [25.2.2015].

HO, D. (2015). Notepad++. <http://notepad-plus-plus.org/> [18.2.2015].

KOSSE, T. (o. J.). FileZilla - The free FTP solution. <https://filezilla-project.org/index.php> [18.2.2015].

MARCOTTE, E. (2010). Responsive Web Design. *A List Apart*. <http://alistapart.com/article/responsive-web-design> [13.2.2015].

MÜLLER, P. (2014). *Einstieg in CSS – Webseiten gestalten mit HTML und CSS*. Bonn: Galileo Press.

MÜNZ, S. (2010). CSS (Cascading Stylesheets) - Webkompetenz. <http://webkompetenz.wikidot.com/html-handbuch:css> [6.4.2015].

MÜNZ, S. (2014). HTML5-Handbuch - Webkompetenz. <http://webkompetenz.wikidot.com/docs:html-handbuch> [6.4.2015].

REFSNES DATA. (o. J.). HTML Tutorial. <http://www.w3schools.com/html/default.asp> [6.4.2015a].

REFSNES DATA. (o. J.). CSS Tutorial. <http://www.w3schools.com/css/default.asp> [6.4.2015b].

WOLF, J. (2016). *HTML5 und CSS3. Das umfassende Handbuch* (Rheinwerk Computing) (1. Aufl., korr. Nachdr.). Bonn: Rheinwerk Verlag.

WORLD WIDE WEB CONSORTIUM (W3C). (2011). Cascading Style Sheets (CSS) Snapshot 2010. <http://www.w3.org/TR/css-2010/> [6.4.2015].

WORLD WIDE WEB CONSORTIUM (W3C). (2014). HTML5. <http://www.w3.org/TR/html5/> [6.4.2015].

	Textvariable	Inhalt
Unterrichtseinheit (UE)		
Bezeichnung	#UE_TITEL!	Septische Granulomatose
Urheberrecht	#UE_RECHT!	Gabriele und Kai Wolff (2015–2016)
Aufgabe		
Bezeichnung	#A_TITEL!	Analyse von Mutationen 2
lfd. Nr. in der UE (Buchstabe möglich)	#A!	C
optional: lfd. Nr. der Teilaufgabe	#T!	2
Anzahl der Hilfekarten zur Aufgabe	#MAX!	7
Hilfekarte		
lfd. Nr. zur (Teil-)Aufgabe	#H!	5
Text der Hilfekarte	#H_TEXT!	Die räumliche Struktur eines Enzyms ergibt sich aus Wechselwirkungen zwischen den Aminosäureresten, die je nach ihren chemischen Eigenschaften Wasserstoffbrücken, unpolare oder polare Wechselwirkungen oder Disulfidbrücken ausbilden können.
optional: Bild	#PI_OPT!	
optional: Überschrift	#PI_TITEL!	
URL	#PI_URL!	
Bildunterschrift	#PI_TEXT!	
Urheberrecht	#PI_RECHT!	<input type="checkbox"/> eigenes Bild oder <input type="checkbox"/> Rechteinhaber:
optional: Video	#VI_OPT!	
optional: Überschrift	#VI_TITEL!	3D-Animation zur Proteinstruktur
lfd. Nr. des Videos in der Hilfekarte	#VI!	1
URL	#VI_URL!	http://www.chemgapedia.de/vsengine/media/vsc/de/video/8/proteinstruktur/proteinstruktur_pur.flv
Videotyp	#VI_TYP!	<input checked="" type="checkbox"/> flv <input type="checkbox"/> mp4 <input type="checkbox"/> ogg <input type="checkbox"/> webm <input type="checkbox"/> Dailymotion <input type="checkbox"/> Vimeo <input type="checkbox"/> YouTube
Höhe des Videos	#VI_H!	480
Breite des Videos	#VI_B!	640
Urheberrecht	#VI_RECHT!	<input type="checkbox"/> eigenes Video oder <input checked="" type="checkbox"/> Rechteinhaber: Wiley-VCH
optional: Audio	#AU_OPT!	
optional: Überschrift	#AU_TITEL!	
optional: Text	#AU_TEXT!	
URL	#AU_URL!	
Audiotyp	#AU_TYP!	<input type="checkbox"/> aac <input type="checkbox"/> mpeg <input type="checkbox"/> ogg <input type="checkbox"/> wav
Urheberrecht	#AU_RECHT!	<input type="checkbox"/> eigene Tonaufnahme oder <input type="checkbox"/> Rechteinhaber:

Tab. 1. Erfassungsmaske für alle Details der Hilfekarte aus Abbildung 5 des Beitrags

Hilfekarten Genetik

Sachtextanalyse



t1p.de/8i2r

Hilfekarten Genetik

Stammbaumanalyse



t1p.de/5aed

Hilfekarten Genetik

Analyse von Mutationen (1)



t1p.de/iqh8

Hilfekarten Genetik

Analyse von Mutationen (2)



t1p.de/1d22

Hilfekarten Genetik
Analyse von Mutationen (3)



t1p.de/mnda

Hilfekarten Waldökologie
Stockwerke des Waldes



t1p.de/jchv

Hilfekarten Waldökologie
Pflanzen und Tiere des Waldes



t1p.de/zwg0