

## Inhaltsverzeichnis

1. Welchen einfachen Texteditor sollte ich zur Nachbearbeitung der erzeugten HTML-Dateien verwenden?.....	2
2. Wie kann ich im Lückentext-Generator Zwischenergebnisse des Textentwurfs sichern, ohne diese erst in die HTML-Datei zu exportieren, in der sich keine neuen Lücken mehr einfügen lassen?.....	3
3. Wie kann ich den Quelltext im Lückentext-Generator schrittweise so optimieren, dass (fast) keine spätere Nachbearbeitung der HTML-Export-Datei notwendig ist?.....	4
4. Wie kann ich auch nachträglich noch weitere mögliche Alternativantworten ergänzen?..	5
5. Bei der Eingabe von Kommazahlen in eine Lücke, werden diese abgeschnitten – wie kann ich das reparieren?.....	6
6. Wie kann ich schmälere Lückeneingabefelder erzeugen, z.B. für die Eingabe von Zahlen oder chemischen Symbolen?.....	7
7. Wie kann ich Formeln (ohne Lücken) in eigenen Absätzen gestalten?.....	8
8. Wie kann ich direkt im Fließtext mathematische Ausdrücke ohne extra Absätze gestalten?.....	9
9. Wie kann ich ausfüllbare Lücken auch innerhalb mathematischer Formeln gestalten ?	10
10. Wie kann ich alle Lösungsmöglichkeiten in zufälliger Reihenfolge am Ende des Lückentexts anzeigbar machen?.....	11
11. Wie kann ich Tabellen mit nachträglich mit Rändern versehen?.....	12
12. Wie kann ich Bilder auch nachträglich einbetten?.....	13
13. Ich möchte den erzeugten Lückentext auch offline nutzen – wie geht das?.....	14
14. Wie kann ich überzählige Zeilenumbrüche entfernen?.....	15
15. Wie kann ich einen Hinweis einbauen, wie bestimmte Sonderzeichen im Lückentext eingegeben werden sollen?.....	16

Diese Zusammenstellung von Antworten zu häufig gestellten Fragen („Frequently Asked Questions“) soll dabei helfen, die Erstellung und Nachbearbeitung von Lückentexten auch mit komplexerem Layout (Tabellen, Listen) sowie mit eingebetteten Bildern zu vereinfachen und bisweilen auftretenden Problemen vorzubeugen.

Beide Editorversionen sind eher für die Erstellung kurzer, einfacher und sprachorientierter Lückentexte optimiert, so dass im Falle mathematischer Formeln und tabellarischer Layouts dann mitunter etwas „Handarbeit“ notwendig ist.

## **1. Welchen einfachen Texteditor sollte ich zur Nachbearbeitung der erzeugten HTML-Dateien verwenden?**

Dieser sollte auf jeden Fall eine farbige Syntaxhervorhebung beherrschen, damit man Fehler beim Bearbeiten schneller erkennt, etwa aus Versehen gelöschte doppelte Hochkommata zur Zeichenkettenbegrenzung oder „defekte“ HTML-Tags.

Als Zeilenumbruch sollte immer Linefeed/\n/Unix/Linux gewählt werden, als Zeichenkodierung UTF-8. Als Einrückungstiefe wird für HTML- und CSS-Abschnitte zwei Leerzeichen empfohlen, für JavaScript-Abschnitte vier Leerzeichen.

Gut geeignet sind die für Linux, Windows und Mac OS X erhältlichen, kostenlosen Open Source Editoren:

- Gedit ([projects.gnome.org/gedit](http://projects.gnome.org/gedit)) und
- Geany ([www.geany.org](http://www.geany.org))

## **2. Wie kann ich im Lückentext-Generator Zwischenergebnisse des Textentwurfs sichern, ohne diese erst in die HTML-Datei zu exportieren, in der sich keine neuen Lücken mehr einfügen lassen?**

Im Lückentexteditor *ohne* direkte HTML-Bearbeitung wird der gesamte Inhalt des Textfelds markiert und dann in ein neues Textdokument in einem einfachen Texteditor hineinkopiert. Dieses Textdokument kann man letztlich z.B. als `lueck_thema_vorlage_stand_2013-02-08.txt` lokal abspeichern.

Im Lückentexteditor *mit* direkter HTML-Bearbeitung dagegen muss zunächst in der zweiten Zeile des dort eingebetteten TinyMCE-Editors ganz rechts der Button mit der Aufschrift „HTML“ angeklickt werden, woraufhin sich ein neues Fenster „HTML-Quellcode bearbeiten“ öffnet. Anschließend wird der gesamte Inhalt des Textfelds in diesem neuen Fenster markiert und dann in ein neues Textdokument in einem einfachen Texteditor hineinkopiert. Dieses Textdokument kann man letztlich z.B. als `lueck_thema_vorlage_stand_2013-02-08.txt` lokal abspeichern.

### **3. Wie kann ich den Quelltext im Lückentext-Generator schrittweise so optimieren, dass (fast) keine spätere Nachbearbeitung der HTML-Export-Datei notwendig ist?**

Klickt man das erste Mal den Knopf „erzeuge Lückentext“ an, so wird in einem neuen Tab der aktuelle Stand des erzeugten HTML-Quelltexts angezeigt.

Diesen Tab lässt man einfach während der weiteren Bearbeitung im ursprünglichen, eigentlichen Lückentext-Generator-Tab weiterhin geöffnet.

Man wechselt zurück zum Lückentext-Generator-Tab und vergleicht den Inhalt des oberen Textfelds mit der unten nun neu angezeigten Vorschau.

Der Inhalt des oberen Textfelds wird nun schrittweise verändert und nach jeder Veränderung drückt man erneut den Knopf „erzeuge Lückentext“. Es wird jeweils im unteren Vorschaubereich kontrolliert, ob die Veränderung den gewünschten Effekt hatte.

Manche Lücken werden in der Vorschau noch mit `#suchbegriff` (beim Lückentext-Generator ohne direkte HTML-Bearbeitung) oder auch mit `#suchbegriff data-mce-bogus="1">` (beim Lückentext-Generator mit direkter HTML-Bearbeitung) angezeigt. Dies liegt häufig an einem fehlenden Leerzeichen vor- oder nach der Lücke (etwa direkt vor einen Zeilenumbruch oder innerhalb eines Tabellenfelds).

Erst wenn alle Lücken in der Vorschau korrekt angezeigt werden, wird ein letztes Mal der Knopf „erzeuge Lückentext“ gedrückt und der jeweils mitaktualisierte Quelltext im HTML-Export -Tab wird lokal als HTML-Datei abgespeichert.

## 4. Wie kann ich auch nachträglich noch weitere mögliche Alternativantworten ergänzen?

Um weitere Alternativantworten zu ermöglichen, öffnet man nach dem HTML-Export die lokal abgespeicherte HTML-Datei mit einem einfachen Texteditor und ergänzt dort z.B.:

```
var alle_loesungen={
  "luecke1": [
    "zu Hause"
  ]
};
```

```
var alle_loesungen={
  "luecke1": [
    "zu Hause",
    "zuhause"
  ]
};
```

**Das Komma zwischen den Alternativantworten darf keinesfalls vergessen werden!**

Für jeden als korrekt einzustufenden Unterschied in der Groß- und Kleinschreibung sowie in der Kombination aus Leerzeichen, Bindestrichen oder Zusammenschreibung muss jeweils eine eigene Antwortvariante angegeben werden!

## 5. Bei der Eingabe von Kommazahlen in eine Lücke, werden diese abgeschnitten – wie kann ich das reparieren?

Da die für normalen Text komfortable Funktion zum Ersetzen von direkt auf die Lücken folgenden Satzzeichen leider nicht 100%ig genau arbeitet, kann es passieren, dass Kommazahlen in Lücken direkt nach dem Komma oder bereits nach einer Stelle hinter dem Komma abgeschnitten werden und direkt nach der Lücke ein Komma (bzw. ein Punkt im Falle englischer Notation) stehenbleibt.

Um dies zu korrigieren, öffnet man nach dem HTML-Export die lokal abgespeicherte HTML-Datei mit einem einfachen Texteditor und ergänzt dort z.B.:

```
var alle_loesungen={
  "luecke1": [
    "3,"
  ]
};
```

zu

```
var alle_loesungen={
  "luecke1": [
    "3,52"
  ]
};
```

Unten im HTML-Quelltext wird dann noch das zusätzliche Komma direkt hinter der Lück

```
<span id="luecke1" class="edit" contenteditable title="">&nbsp;</span> ,
```

entfernt:

```
<span id="luecke1" class="edit" contenteditable title="">&nbsp;</span>
```

## **6. Wie kann ich schmälere Lückeneingabefelder erzeugen, z.B. für die Eingabe von Zahlen oder chemischen Symbolen?**

In der auf die Lösungssammlung direkt folgende Zeile des exportieren HTML-Quelltextes

```
var eingabefeldbreite='80px'; //Für Wörter ca. 80px, bei Berechnungen z.B. 40px
```

kann man mit einem einfachen Texteditor den Wert von 80 Pixeln für Wörter auf z.B. 40 Pixel für Zahlen verringern:

```
var eingabefeldbreite='40px'; //Für Wörter ca. 80px, bei Berechnungen z.B. 40px
```

Dadurch werden alle Lücken innerhalb dieses Lückentextdokuments dann einheitlich entsprechend verschmälert.

## 7. Wie kann ich Formeln (ohne Lücken) in eigenen Absätzen gestalten?

Die Math-Umgebung wird in diesem Fall mit `\[` eingeleitet und mit `\]` beendet.

Ein Beispielquelltextausschnitt wäre also für den Satz von Pythagoras:

```
\[ a^2 + b^2 = c^2 \]
```

Dies würde dann später mit Hilfe von MathJaX dargestellt in der Form:

$$a^2 + b^2 = c^2$$

## 8. Wie kann ich direkt im Fließtext mathematische Ausdrücke ohne extra Absätze gestalten?

Die Math-Umgebung wird in diesem Fall mit `\(` eingeleitet und mit `\)` beendet.

Ein Beispielquelltextausschnitt für diesen Fall wäre für eine Wurzel:

Die `\( \sqrt{4} \)` ist gleich `\( 2 \)`.

Dies würde später mit Hilfe von MathJaX dargestellt in der Form:

Die  $\sqrt{4}$  ist gleich 2 .

## 9. Wie kann ich ausfüllbare Lücken auch innerhalb mathematischer Formeln gestalten ?

Im Falle von Brüchen können diese auch mit der Lückentext-Generator Version mit direkter HTML-Bearbeitung ([quizdidaktik.de/lueckedit2](http://quizdidaktik.de/lueckedit2)) als dreizeilige Tabelle erstellt werden in der Form:

A		C
—	=	—
B		#D

Der dazugehörige Quelltext könnte (mit unsichtbaren Umrandungen) etwa lauten:

```
<table>
  <tr><td>A</td><td></td>&nbsp;<td>C</td></tr>
  <tr><td><hr></td><td></td>=<td><hr></td></tr>
  <tr><td>B</td><td></td>&nbsp;<td> #D </td></tr>
</table>
```

Einzeilige Schreibweisen ohne komplexe Positionierung können auch mit Hilfe der entsprechenden Unicode-Zeichen erzeugt werden. Übersichten dazu findet man u.a. auf:

[de.wikipedia.org/wiki/Mathematische\\_Zeichen\\_in\\_Unicode](http://de.wikipedia.org/wiki/Mathematische_Zeichen_in_Unicode)

[www.kapslog.de/wissen/2012/mathe-symbole-unicode](http://www.kapslog.de/wissen/2012/mathe-symbole-unicode)

[de.selfhtml.org/html/referenz/zeichen.htm](http://de.selfhtml.org/html/referenz/zeichen.htm)

[www.mathe-online.at/formeln](http://www.mathe-online.at/formeln)

[www.cilie.org/esa/information/mathematische-zeichen.htm](http://www.cilie.org/esa/information/mathematische-zeichen.htm)

## 10. Wie kann ich alle Lösungsmöglichkeiten in zufälliger Reihenfolge am Ende des Lückentexts anzeigbar machen?

In der sechstletzten Zeile der erzeugten HTML-Datei

```
<input type="button" id="loes" value="Zeige Lösungsmöglichkeiten"
onclick="zeige_loesungswörter();" hidden="hidden" />
```

wird einfach mit einem einfachen Texteditor die letzte Zuweisung `hidden="hidden"` entfernt, so dass übrig bleibt:

```
<input type="button" id="loes" value="Zeige Lösungsmöglichkeiten"
onclick="zeige_loesungswörter();" />
```

Lädt man nun dieses HTML-Dokument nach dem Speichern erneut im Webbrowser, so erhält man einen zusätzlichen Knopf mit der Aufschrift „Zeige Lösungsmöglichkeiten“.

## 11. Wie kann ich Tabellen nachträglich mit Rändern versehen?

Um alle Tabellen mit Umrandungen anzuzeigen, öffnet man nach dem HTML-Export die lokal abgespeicherte HTML-Datei mit einem einfachen Texteditor und ergänzt dort im Style-Abschnitt z.B.:

```
table, td {
  border-collapse:collapse;
  border-style:solid;
  border-width:1px;
  border-color:grey;
}
td {
  padding:5px;
}
```

Sollen aber bei bestimmten Tabellen weiterhin keine Umrandungen angezeigt werden, so fügt man zusätzlich im Style-Abschnitt hinzu:

```
table.ohne_linie td {
  border-color:white;
  padding:0px;
}
```

Bei der betreffenden Tabelle muss dann noch unten im Body-Abschnitt ergänzt werden:

```
<table>
  <tr>...
```

zu

```
<table class="ohne_linie">
  <tr>
```

## 12. Wie kann ich Bilder auch nachträglich einbetten?

Um zusätzliche Bilder einzufügen, öffnet man nach dem HTML-Export die lokal abgespeicherte HTML-Datei mit einem einfachen Texteditor und ergänzt dort an der entsprechenden Stelle im Body-Abschnitt.

### Variante A:

Liegt die Bilddatei an einem bekannten (und online erreichbaren) Speicherort, so kann man sie (sofern dies lizenzrechtlich und bezüglich des Urheberrechts erlaubt ist) einbinden in der Form:

```

```

### Variante B:

Man kann zunächst mit Hilfe des Programms Bild2HTML ([quizdidaktik.de/bild2html](http://quizdidaktik.de/bild2html)) zuerst das Bild in eine DataURL umwandeln und diese dann dem Image-Tag als Bildquelle zuweisen.

Das gesamte (sehr lange) Image-Tag kann dann an die entsprechende Stelle im Body-Abschnitt kopiert werden.

Der Vorteil dieser Variante besteht darin, dass das Bild selbst Teil der Lückentext-HTML-Datei ist und somit immer verfügbar ist, unabhängig vom Speicherort.

Der Vorteil des erst nachträglichen Einfügens dieser bereits bei kleinen Bildern (selbst kleiner als 300x300 Pixel) sehr langen Textzeile liegt in der davor weitaus flüssigeren Bearbeitbarkeit der Zwischenergebnisse, da sowohl der Webbrowser beim Erzeugen der HTML-Datei als auch die Texteditoren danach sehr träge reagieren und im Extremfalls sich sogar „aufhängen“ können, wodurch dann im Falle einer fehlenden vorherigen Sicherung sogar Zwischenergebnisse verloren gehen können!

## 13. Ich möchte den erzeugten Lückentext auch offline nutzen – wie geht das?

Falls man keine mathematischen Formeln mit MathJaX benutzen möchte, so kann man die folgenden Zeilen wie folgt verändern:

```
<script type="text/x-mathjax-config">MathJax.Hub.Config({TeX: {extensions: ["mhchem.js"]}});</script>
  <script type="text/javascript"
src="http://cdn.mathjax.org/mathjax/latest/MathJax.js?config=TeX-AMS-MML_HTMLorMML"></script>
```

in:

```
<!-- <script type="text/x-mathjax-config">MathJax.Hub.Config({TeX: {extensions: ["mhchem.js"]}});</script>
  <script type="text/javascript"
src="http://cdn.mathjax.org/mathjax/latest/MathJax.js?config=TeX-AMS-MML_HTMLorMML"></script> -->
```

Fortgeschrittene können natürlich auch eine offline-Version von MathJaX verfügbar machen. Weitere Hinweise dazu findet man unter [docs.mathjax.org/en/latest/installation.html](http://docs.mathjax.org/en/latest/installation.html)

Solange man noch online ist, muss nach noch von der Seite <http://ajax.googleapis.com/ajax/libs/jquery/1.8.3/jquery.min.js>

die Datei `jquery.min.js` herunterladen und direkt neben den Lückentext speichern. Nun ändert man die Zeile

```
  <script type="text/javascript"
src="http://ajax.googleapis.com/ajax/libs/jquery/1.8.3/jquery.min.js"></script>
```

ab in:

```
<script type="text/javascript" src="jquery.min.js"></script>
```

Fertig – sobald man nun die Lückentext-HTML-Datei öffnet, kann man diese auch im Offline-Modus normal bearbeiten.

## 14. Wie kann ich überzählige Zeilenumbrüche entfernen?

Um überzählige Zeilenumbrüche (diese machen sich z.B. durch zu große Zeilenabstände bemerkbar) zu entfernen, öffnet man nach dem HTML-Export die lokal abgespeicherte HTML-Datei mit einem einfachen Texteditor und wählt unter

„Suchen“ > „Ersetzen“  
dann im neuen Fenster eingeben  
Suchen nach: `<br />`  
Ersetzen durch:

Anschließend kann man über abwechselndes „Suchen“ und „Ersetzen“ alle überzähligen Zeilenumbrüche (`<br />`) entfernen. Insbesondere innerhalb eines Tabellengerüsts führen diese sonst zu einem zu großen Abstand zum vorherigen Seiteninhalt.

## 15. Wie kann ich einen Hinweis einbauen, wie bestimmte Sonderzeichen im Lückentext eingegeben werden sollen?

Sonderzeichen mit Akzenten können beim späteren Bearbeiten der Lückenfelder eingegeben werden mit...

...einer **Umstellung des Tastaturlayouts** auf die gewünschte Zielsprache

...der **Unicode-Eingabemethode** des jeweiligen Betriebssystems bzw. der Desktopumgebung, dazu siehe auch [de.wikipedia.org/wiki/Unicode#Eingabemethoden](http://de.wikipedia.org/wiki/Unicode#Eingabemethoden)

...der **Sonderzeichentastatur des Lückentext-Generators** selbst. Die damit erzeugten Zeichen müssen aber in diesem Fall anschließend mit „kopieren“ & „einfügen“ (oder Strg+C & Strg+V) dann an die richtige Stelle der gewählten Lücke hineinkopiert werden.

Um dem Lückentext eine Kurzanleitung für diesen letzten Fall beizufügen, kann die HTML-Datei mit einem einfachen Texteditor geöffnet werden und anschließend vor der Zeile

```
<input type="button" value="Überprüfen" onclick="ueberpruefen();" />
```

der folgende Abschnitt hinzugefügt werden:

```
<div style="color:#696969;font-size:10px;">
```

```
Hinweis: Falls <b>Akzente</b> oder andere Sonderzeichen wie das ñ nicht korrekt über die Tastatur eingegeben werden können, so kann man diese in einem neuen Browsertab im
```

```
<a class="linkbereich"
```

```
  href="http://quizdidaktik.de/lueckedit/index.html"
```

```
  target="_blank"
```

```
  style="color:#696969;">Lückentext-Generator</a>
```

```
mit Hilfe der dortigen <b>Sonderzeichentastatur</b> zunächst dort in ein beliebiges Feld eingeben und anschließend an die entsprechende Stelle im gesuchten Wort auf dieser Lückentextseite hineinkopieren.
```

```
</div>
```

```
<input type="button" value="Überprüfen" onclick="ueberpruefen();" />
```