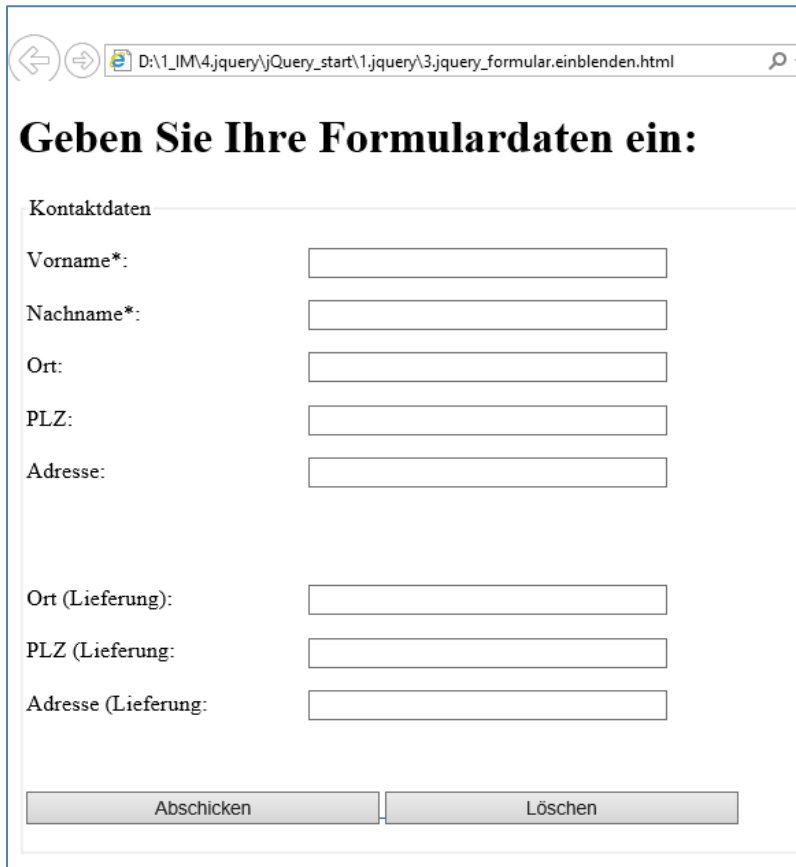


jQuery: Bestellformular mit einblendbarer Lieferadresse

Erstelle ein kleines Bestellformular für einen Online-Shop. Manchmal weichen die Rechnungs- und Lieferadresse voneinander ab. Damit man dem Kunden nicht unnötige Eingabefelder präsentiert, sollen die Felder für die Lieferadresse nur dann angezeigt werden, wenn es der Käufer explizit wünscht.

Erstelle folgende HTML und CSS-Datei:



Geben Sie Ihre Formulareinblendungsdaten ein:

Kontaktdaten

Vorname*:

Nachname*:

Ort:

PLZ:

Adresse:

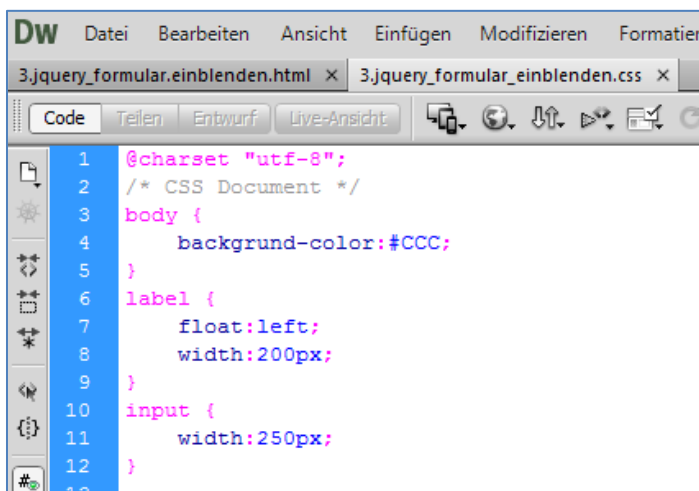
Ort (Lieferung):

PLZ (Lieferung):

Adresse (Lieferung):

Abschicken Löschen

CSS:



```
1 @charset "utf-8";
2 /* CSS Document */
3 body {
4     background-color:#CCC;
5 }
6 label {
7     float:left;
8     width:200px;
9 }
10 input {
11     width:250px;
12 }
13
```

```
3.jquery_formular.einblenden.html* x 3.jquery_formular_einblenden.css x
Quellcode jquery-3.0.0.js 3.jquery_formular_einblenden.css
Code Teilen Entwurf Live-Ansicht
Titel:

1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="utf-8">
5     <script src="js/jquery-3.0.0.js">
6   </script>
7     <link rel="stylesheet" href="css/3.jquery_formular_einblenden.css">
8   </head>
9   <body>
10    <h1>Geben Sie Ihre Formular Daten ein:</h1>
11    <form action="" method="post">
12      <fieldset>
13        <legend>Kontaktdaten</legend>
14        <div id="Rechnungsadresse">
15          <p><label for="Vorname">Vorname*: </label>
16          <input name="vorname" type="text" required</p>
17          <p><label for="Nachname">Nachname*:</label>
18          <input name="nachname" type="text" required</p>
19          <p><label for="Ort">Ort: </label>
20          <input name="ort" type="text"></p>
21          <p><label for="plz">PLZ: </label>
22          <input name="plz" type="text"></p>
23          <p><label for="Adresse">Adresse: </label>
24          <input name="adresse" type="text"></p>
25        </div>
26        <br><br>
27        <div id="Lieferadresse">
28          <p><label for="Ort_L">Ort (Lieferung): </label>
29          <input name="ort_lief" type="text">
30          </p>
31          <p><label for="plz_L">PLZ (Lieferung: </label>
32          <input name="plz_lief" type="text"></p>
33          <p><label for="Adresse_L">Adresse (Lieferung: </label>
34          <input name="adresse_lief" type="text"></p>
35          <br>
36        </div>
37        <br>
38        <input type="submit" value="Abschicken">
39        <input type="reset" value="Löschen">
40      </fieldset>
41    </form>
42  </body>
43 </html>
44
```

a) Link einfügen: (inkl. ID einfügen, um später besser zugreifen zu können)

- 1.) Beim ersten Aufruf der Seite darf der DIV-Container „Lieferadresse“ nicht angezeigt werden.
- 2.) Link einfügen, der durch Anklick erst den Container „Lieferadresse“ anzeigt.

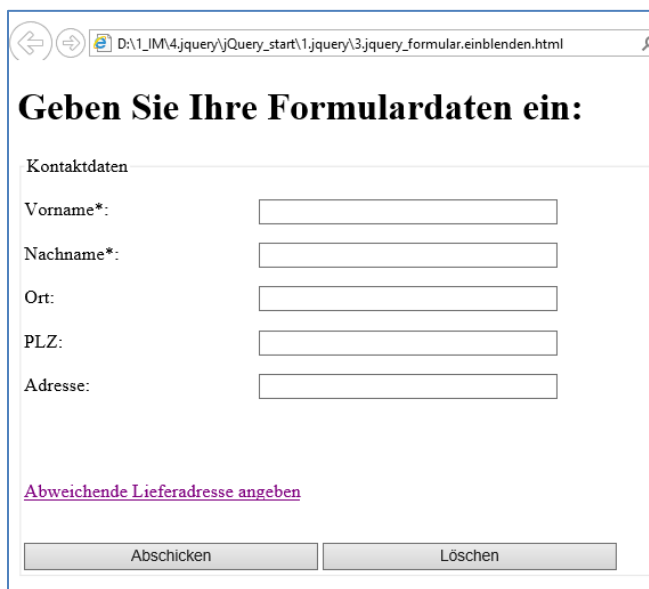
Daher benötigt man zuerst einen Link, der angeklickt werden kann. Erstelle daher am Ende, nach dem letzten <div> einen <a href> - Link. Dieser soll die ID für den Container „Lieferadresse“ beinhalten, damit nicht alle <a> angesprochen werden, sondern nur dieser eine Container.

```
<input name="adresse_lief" type="text"></p>
<br>
</div>
<p><a id="LieferLink" href="javascript:void(0);">Abweichende Lieferadresse angeben</a></p>
<br>
<input type="submit" value="Abschicken">
<input type="reset" value="Löschen">
</div>
```

b) jQuery: hide()

Nun folgt die die Funktion, die mit der jQuery-Methode „hide()“ den entsprechenden Container ausblendet.

```
41     </fieldset>
42 </form>
43 <script>
44     $(document).ready(function() {
45         $("#Lieferadresse").hide();
46     });
47 </script>
48 </body>
49 </html>
```



The screenshot shows a web browser window with the address bar containing the file path: D:\1_JIM\4_jquery\jQuery_start\1_jquery\3_jquery_formular_einblenden.html. The page title is "Geben Sie Ihre Formularein:". Below the title, there is a section titled "Kontaktdaten" with five input fields: "Vorname*", "Nachname*", "Ort:", "PLZ:", and "Adresse:". Below these fields, there is a link "Abweichende Lieferadresse angeben" which is currently hidden. At the bottom of the form, there are two buttons: "Abschicken" and "Löschen".

c)Methode „toggle()“

Dieser Container ist jetzt ausgeblendet. Aber er soll wieder eingeblendet werden, wenn man auf den Link drückt. Daher verwende die jQuery-Methode „toggle()“.

Diese ermittelt automatisch den aktiven Anzeigestatus des selektierten Elements und löst anschließend die Methoden hide() oder show() aus.

Darunter muss man noch auf den Link mit der ID=“LieferLink“ eine Klick-Funktion legen, die die obige „Toggle-Funktion“ ausführt.

```
41     </fieldset>
42 </form>
43 <script>
44     $(document).ready(function() {
45         $("#Lieferadresse").hide();
46         function zeigeLieferadresse() {
47             $("#Lieferadresse").toggle();
48         }
49         $("#LieferLink").click(zeigeLieferadresse);
50     });
51 </script>
52 </body>
53 </html>
```

Ergebnis: Nun wird der Container, der ursprünglich ausgeblendet ist, jederzeit per Klick auf den Link wieder eingeblendet.

Geben Sie Ihre Formulardaten ein:

Kontaktdaten

Vorname*:

Nachname*:

Ort:

PLZ:

Adresse:

Ort (Lieferung):

PLZ (Lieferung):

Adresse (Lieferung):

[Abweichende Lieferadresse angeben](#)

d)Text des Links nach dem Anklick ändern

Wird auf den Link „Abweichende Lieferadresse angeben“ geklickt sollte sich der Text ändern in „Sie haben auf den Link geklickt“.

Somit kennt sich der Besucher aus und weiß, dass der Link bereits besucht wurde.

Die vorher angelegte ID des Links (id=“LieferLink“) eignet sich nun sehr gut dafür, angesprochen zu werden. Danach wird mit der

Methode: text()

ein anderer Inhalt zugewiesen. Nämlich genau auf die Zeile des Links „#LieferLink“.

```
<script>
$(document).ready(function() {
  $("#Lieferadresse").hide();
  function zeigeLieferadresse() {
    $("#Lieferadresse").toggle();
    var text = $("#Lieferadresse").text(); //Liefert den aktuellen Text zurück
    $("#LieferLink").text("Sie haben auf den Link geklickt"); //Überschreiben des Textes
  }
  $("#LieferLink").click(zeigeLieferadresse);
});
</script>
```

Ergebnis: Nach dem Klick ist der Text geändert:

Ort (Lieferung):	<input type="text"/>
PLZ (Lieferung):	<input type="text"/>
Adresse (Lieferung):	<input type="text"/>
Sie haben auf den Link geklickt	
<input type="button" value="Abschicken"/> <input type="button" value="Lösch"/>	

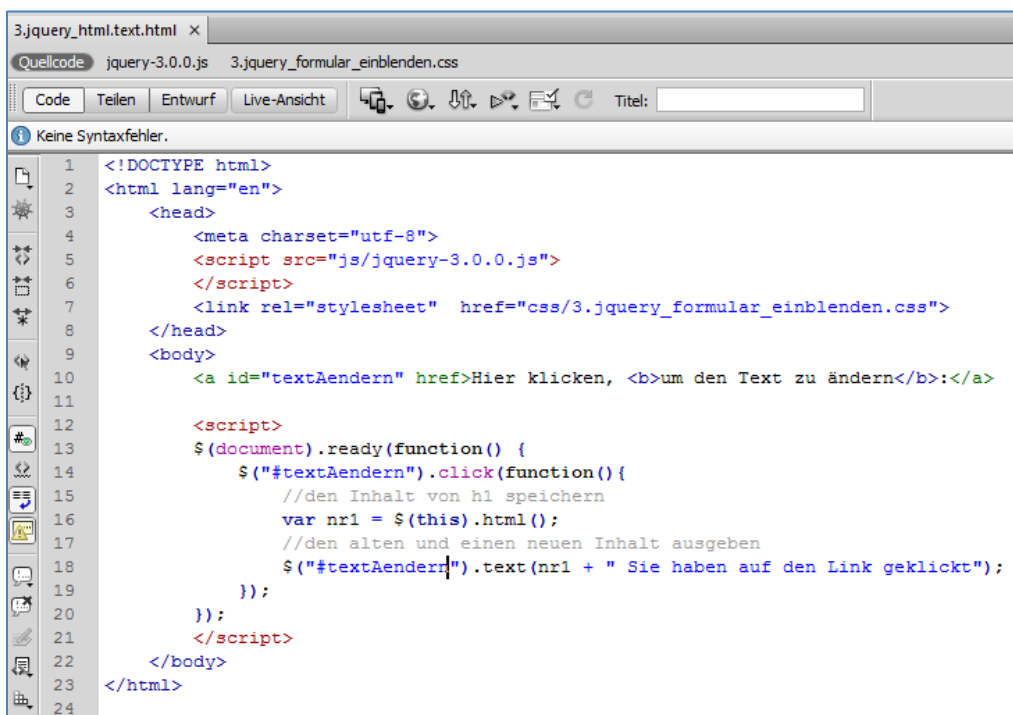
Die Methode „text()“ kann den Text sowohl zurückliefern als auch ändern. Somit kann man mit diesem einen Befehl den Textinhalt des selektierten Elements z.B. für eine neue Variable ermitteln als auch direkt verändern. Man kann jedoch nicht mit dieser Methode auf ein „input“-Element anwenden!

Theorie: Elemente verändern:

html() liefert den Inhalt eines HTML-Tags zurück
htm(var) ersetzt den Inhalt eines HTML-Tags
text() liefert den Inhalt eines HTML-Tags zurück
text(var) ersetzt den Inhalt eines HTML-Tags

Text(var) gibt den Inhalt enthält einen reinen Textstring und gibt somit den Inhalt „escaped“ zurück, d.h. mit allen Codes wie z.B. „Klicken Sie hier.“

Übung: erstelle eine Datei „3.jquery_html.text.html“ mit folgendem Code:

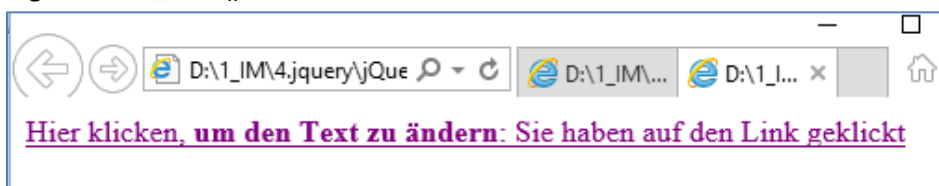


```
1 <!DOCTYPE html>
2 <html lang="en">
3   <head>
4     <meta charset="utf-8">
5     <script src="js/jquery-3.0.0.js">
6   </script>
7     <link rel="stylesheet" href="css/3.jquery_formular_einblenden.css">
8   </head>
9   <body>
10    <a id="textAendern" href="Hier klicken, <b>um den Text zu ändern</b>:</a>
11
12    <script>
13      $(document).ready(function() {
14        $("#textAendern").click(function() {
15          //den Inhalt von h1 speichern
16          var nr1 = $(this).html();
17          //den alten und einen neuen Inhalt ausgeben
18          $("#textAendern").text(nr1 + " Sie haben auf den Link geklickt");
19        });
20      });
21    </script>
22  </body>
23 </html>
24
```

Ergebnis bei „text()“:



Ergebnis bei „html()“: ohne Sonder-Zeichen



Mit der Methode „click()“ belegt man die ID=“textAendern“ innerhalb des Links mit einem onclick-Handler mit einer neuen Funktion, die den Inhalt des angeklickten Elements mit Hilfe der Methode „html()“ in die Variable „var nr1“ überträgt. Im zweiten Schritt gibt man diesen gespeicherten Inhalt

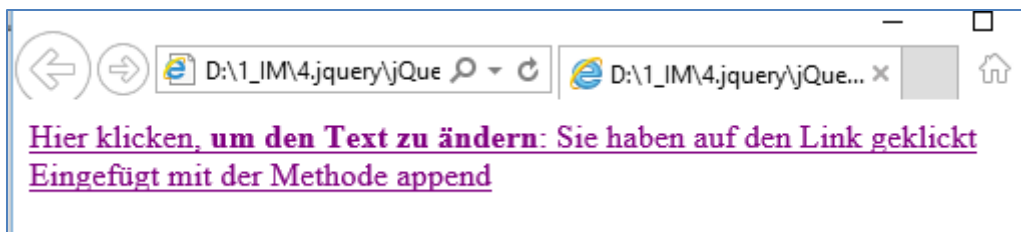
wieder an das angeklickte Element zurück, entweder mit „html(var)“ (hier: html(nr1)) oder „text(var)“. Das Ergebnis mit Hilfe der Anwendung „html()“ ist besser und schöner. „text()“ gibt einen reinen Textstring aus der auch die Sonderzeichen enthält.

Methode „append()“

Mit dieser Methode lassen sich einzelne Inhalte oder auch ganze Elemente innerhalb eines selektierten Elements einfügen. Grundsätzlich werden die zusätzlichen Inhalte am Ende eingefügt.

Übung:

Füge im oberen Beispiel den Text „Eingefügt mit der Methode append“ ein. Es soll auch ein Zeilenumbruch davor die Ausgabe verschönern:



```
<script>
$(document).ready(function() {
  $("#textAendern").click(function() {
    //den Inhalt von h1 speichern
    var nr1 = $(this).html();
    //den alten und einen neuen Inhalt ausgeben
    $("#textAendern").html(nr1 + " Sie haben auf den Link geklickt");
    $(this).append("<br>Eingefügt mit der Methode append");
  });
});
</script>
</body>
```

Tipp.: mit der Methode „prepend()“ wird der Inhalt am Anfang eingefügt.

Quelle:

Sascha Schoppengerd, in: jQuery Praxiseinstieg, 2010, mitp, Heidelberg, S.57-77