Ein Buch zum Mitmachen und Verstehen

jQuery

von Kopf bis Fuß

Erstellen Sie Web-Applikations ohne nervige Brower-Plugins

Implementieren Sie komplexe HTML-Formulare in kürzester Zeit

Schreiben Sie Ihre eigenen maßgeschneiderten jQuery-Funktionen

Machen Sie Ihre Webseite durch wenige Code-Zeilen interaktiv

Animieren Sie Ihre Webseiten

O'REILLY®

Ryan Benedetti & Ronan Cranley
Deutsche Übersetzung von Jørgen W. Lang

Ryan Benedetti / Ronan Cranley, jQuery von Kopf bis Fuß, O´Reilly, ISBN 9783868991895

Inhaltsverzeichnis (Zusammenfassung)

	Einführung	xxiii
1	jQuery: Die ersten Schritte, *Webpage-Action*	1
2	Selectors and Methods: *Grab and Go*	35
3	jQuery Events and Functions: *Making Things Happen on Your Page*	75
4	jQuery Web Page Manipulation: *Mod the DOM*	123
5	jQuery Effects and Animation: *A Little Glide in Your Stride*	175
6	jQuery and JavaScript: ~~*Luke,*~~ *jQuery, I Am Your Father!*	215
7	Custom Functions for Custom Effects: *What Have You Done for Me Lately?*	253
8	jQuery and Ajax: *Please Pass the Data*	291
9	Handling JSON Data: *Client, Meet Server*	325
10	jQuery UI: *Extreme Form Makeover*	371
11	jQuery and APIs: *Objects, Objects Everywhere*	411
i	Leftovers: *The Top Ten Things (We Didn't Cover)*	447
ii	Set Up a Development Environment: *Get Ready for the Big Times*	461

Inhaltsverzeichnis (jetzt aber richtig)

Einführung

Ihr Hirn und jQuery. Da sitzen Sie nun und versuchen, etwas zu lernen. Gleichzeitig will Ihr Hirn Ihnen einen Gefallen tun, indem es dafür sorgt, dass vom Gelernten nichts hängen bleibt. Ihr Hirn denkt: »Besser Platz für wichtige Sachen lassen, zum Beispiel, welche wilden Tiere man besser vermeidet und ob man nackt Snowboarden sollte.« Wie können Sie Ihr Hirn davon überzeugen, dass Ihr Leben davon abhängt, jQuery zu können?

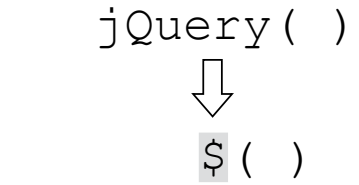
Für wen ist dieses Buch gedacht?	xxii
Metakognition: Nachdenken übers Denken	xxv
Machen Sie sich Ihr Gehirn untertan	xxvii
Lies mich	xxviii
Die technischen Gutachter	xxxii
Danksagungen	xxxiii

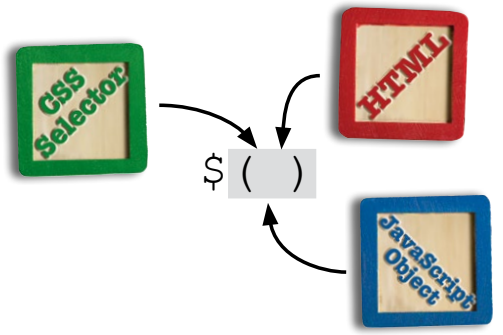
jQuery: Die ersten Schritte

1 Webpage-Action

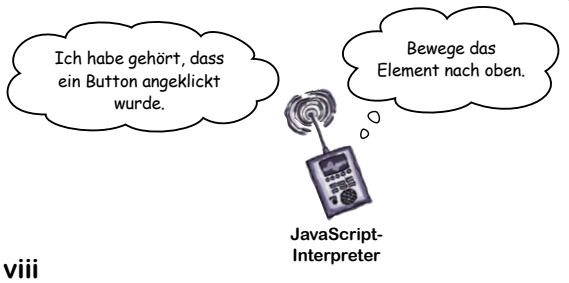
Ihre Webseiten sollen mehr können.

Mit HTML und CSS kennen Sie sich schon recht gut aus. Jetzt wollen Sie auch die Skriptprogrammierung erlernen. Dabei wollen Sie aber nicht den Rest Ihres Lebens mit dem Schreiben von Code verschwenden. Sie brauchen eine Skriptbibliothek, mit der Sie Webseiten ganz nach Bedarf verändern können. Und weil wir uns gerade etwas wünschen, soll sie am besten auch noch mit AJAX und PHP zusammenarbeiten. Außerdem sollten drei Zeilen Code ausreichen, wo andere clientseitigen Sprachen fünfzehn brauchen. Reines Wunschdenken, meinen Sie? Ganz und gar nicht! Es wird Zeit, dass Sie jQuery kennenlernen.





Webseiten mit mehr »Wumms«	2
HTML und CSS sind OK, aber …	3
… möge das Skript mit Ihnen sein	4
Willkommen bei jQuery (und JavaScript)!	5
Ein Blick in den Browser	7
Die versteckte Struktur einer Webseite	8
jQuery macht das DOM einfacher	9
Wie geht *das* denn?	11
jQuery wählt Elemente genauso aus wie in CSS	13
(Aus-)Wahlverwandtschaften: Stil und Skript	14
jQuery-Selektoren, stets zu Diensten	15
jQuery in der Übersetzung	16
Ihre erste jQuery-Aufgabe	20
Die HTML- und CSS-Dateien vorbereiten	24
Immer herein …	26
Möge der Fade mit Ihnen sein	27
Sie haben die Kampagne gerettet. Bravo!	30
Ihr jQuery-Werkzeugkasten	33



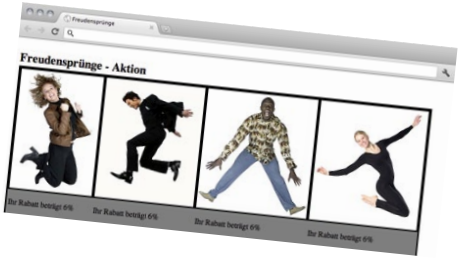
Ryan Benedetti / Ronan Cranley, jQuery von Kopf bis Fuß, O´Reilly, ISBN 9783868991895

2 Selektoren und Methoden

Einpacken und Mitnehmen

jQuery hilft Ihnen, Elemente einer Webseite auszuwählen

und alles mögliche damit anzustellen. In diesem Kapitel beschäftigen wir uns mit jQuery-Selektoren. Wir schnappen uns Elemente einer Seite und benutzen Methoden, um bestimmte Dinge mit den Elementen anzustellen. Wie ein riesiges Buch voller Zaubersprüche hilft uns die jQuery-Bibliothek, tonnenweise Sachen bei Bedarf zu verändern. Bilder können verschwinden und wieder auftauchen. Wir können einen bestimmten Textteil auswählen und seine Schriftgröße animieren. Und damit genug geredet. Suchen Sie sich ein paar Elemente aus und los geht's!

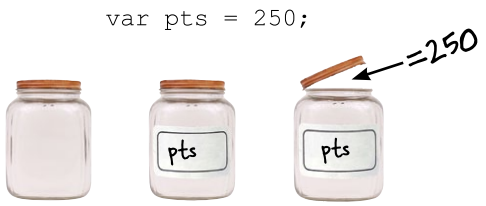


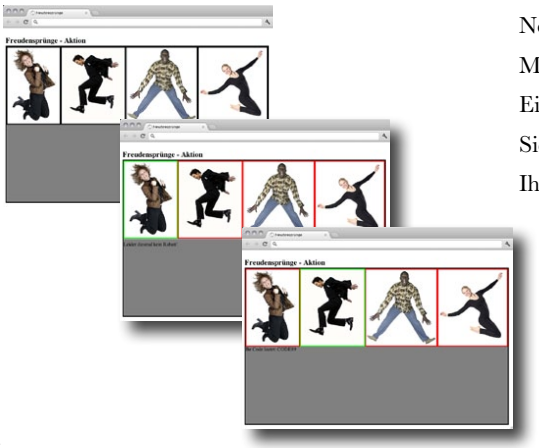
Freudensprünge	36
Was sind die Projektanforderungen?	37
div-ertimento	39
Ein Click-Event unter der Lupe	42
Bauen Sie die click-Methode in Ihre Seite ein	45
Werden Sie spezifischer	47
Elemente mit Klasse	48
Elemente ID-entifizieren	49
Die Webseite verdrahten	52
Gleichzeitig auf unserer Liste	55
Speicherplatz anlegen	56
Verschiedene Dinge verketten	57
Zur selben Zeit im Code …	58
Die Nachricht mit der append-Funktion in die Seite einfügen	59
Das funktioniert schon ganz gut, aber …	61
Gib mir $(this)	63
$(this) bei der Arbeit	64
Gutes Gelingen mit remove	66
Nachkommen-Selektoren für mehr Tiefe	67
Jetzt sind Sie dran, Freudensprünge zu machen	73
Ihr jQuery-Werkzeugkasten	74

jQuery-Events und -Funktionen

3 Dinge auf Ihrer Seite geschehen lassen

Mit jQuery ist es einfach, eine Webseite mit Action und Interaktivität zu versehen. In diesem Kapitel zeigen wir Ihnen, wie Sie Ihre Seite dazu bringen, auf Aktionen von Leuten zu reagieren. Indem Ihr Code auf die Aktionen der Benutzer reagiert, kann das Niveau Ihrer Website deutlich steigen. Außerdem sehen wir uns an, wie Sie wiederwendbare Funktionen erstellen können. Dadurch brauchen Sie den Code nur einmal zu schreiben, können ihn aber beliebig oft benutzen.





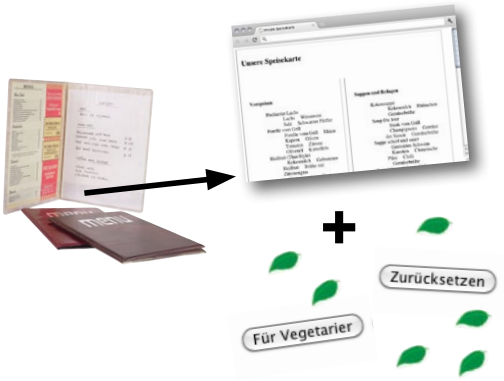
Ihre jQuery-Fähigkeiten sind wieder einmal gefragt	76
Der Geldmensch hat irgendwie recht ...	77
Ereignisreiche Seiten	79
Hinter den Kulissen eines Event-Listeners	80
Ein Event binden	81
Events auslösen	82
Ein Event entfernen	86
Die Elemente der Reihe nach durchgehen	90
Ihre Projektstruktur	96
Die Dinge funktions-fähig machen	100
Die Innereien einer Funktion	101
Anonyme Funktionen	102
Benannte Funktionen als Event-Handler	103
Variablen an eine Funktion übergeben	106
Funktionen können auch Werte zurückgeben	107
Benutzen Sie Bedingungen, um Entscheidungen zu treffen	109
Noch mehr Hilfe für die Freudensprünge-Aktion	113
Methoden zum Ändern von CSS	115
Ein hover-Event hinzufügen	117
Sie haben es fast geschafft ...	119
Ihr jQuery-Werkzeugkasten	122

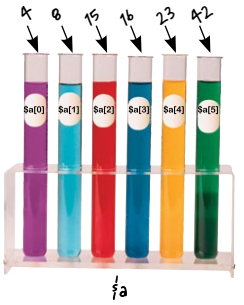
Ryan Benedetti / Ronan Cranley, jQuery von Kopf bis Fuß, O´Reilly, ISBN 9783868991895

Webseiten mit jQuery manipulieren

4 Mod the DOM

Dass die Seite fertig geladen ist, heißt nicht, dass sie deshalb die gleiche Struktur behalten muss. In Kapitel 1 haben wir bereits gesehen, wie beim Laden einer Seite das DOM aufgebaut wird, um die Struktur der Seite zu erstellen. In diesem Kapitel wollen wir sehen, wie man sich in der DOM-Struktur auf- und abbewegt. Wir werden Ihnen zeigen, wie man mit der Hierarchie der Elemente und mit Eltern-Kind-Beziehungen arbeitet, um die Seitenstruktur nach Bedarf mit jQuery zu verändern.





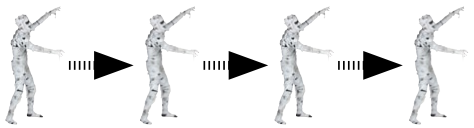
Das Webville-Restaurant hätte gerne eine interkative Speisekarte	124
Die vegetarische Version	125
Elemente mit Klasse	130
Knöpfen Sie sich die Dinge vor	133
Was kommt jetzt?	135
Im DOM-Baum herumklettern	140
Traversal-Methoden bewegen sich im DOM	141
Methoden verketten, um weiter zu klettern	142
Variablen können auch Elemente speichern	149
Schon wieder dieses Dollarzeichen	150
Die Speichermöglichkeiten mit Arrays erweitern	151
Elemente in einem Array speichern	152
Elemente mit replaceWith verändern	154
Wie kann replaceWith uns helfen?	155
Denken Sie voraus, bevor Sie replaceWith benutzen	157
replaceWith funktioniert nicht in jeder Situation	158
HTML-Inhalte in das DOM einfügen	159
Die Auswahl mit Filtermethoden einschränken (Teil 1)	161
Die Auswahl mit Filtermethoden einschränken (Teil 2)	162
Der Hamburger kommt zurück	165
Wo ist das Rind (ähh … Fleisch)?	166
Ein fleischiges Array	167
Die each-Methode kann eine Schleife über ein Array ausführen	168
Das war's … oder?	171
Ihr jQuery-Werkzeugkasten	174

jQuery-Effekte und -Animationen

5 Vom Gleiten und Schreiten

Die Fähigkeit, Dinge auf Ihrer Webseite passieren zu lassen, ist schön und gut, aber wenn Sie es nicht schaffen, das Ganze auch noch cool aussehen zu lassen, werden die Leute Ihre Site nicht benutzen wollen. Und genau da kommen jQuery-Effekte und -Animationen ins Spiel. In diesem Kapitel lernen Sie, wie Sie Elemente mit fließenden Übergängen versehen können, bestimmte wichtige Teile von Elementen anzeigen oder verstecken können, oder ein Elemente vergrößern und verkleinern können – und das alles vor den Augen Ihrer Benutzer. Sie werden außerdem sehen, wie Sie Animationen in bestimmten zeitlichen Abständen ablaufen lassen, um Ihrer Seite ein sehr dynamisches Aussehen zu geben.

DoodleStuff braucht eine Web-Applikation	176
Werden Sie zum Monstermacher	177
Der Monstermacher braucht ein Layout und Positionierung	178
Etwas mehr Struktur und Stil	181
Die Schnittstelle anklickbar machen	182
Den Blitz-Effekt umsetzen	187
Wie animiert jQuery Elemente?	188
Ein- und Ausblendeffekte animieren die CSS-Eigenschaft opacity	189
Bei den slide-Funktionen geht es um die Höhe	190
Die Fade-Effekte anwenden	192
Effekt-Methoden miteinander verketten	193
Funktionen zeitgesteuert wiederholen	194
Bauen Sie die Blitz-Funktionen in Ihr Skript ein	197
Selbstgemachte Effekte mit animate	199
Was animiert werden kann und was nicht	200
animate Stile nach und nach	202
Von wo nach wo genau?	205
Elemente absolut und relativ bewegen	206
Dinge durch die Kombination von Operatoren relativ bewegen	207
Bauen Sie die animate-Funktion in Ihr Skript ein	209
Und ganz ohne Flash!	212
Ihr jQuery-Werkzeugkasten	214



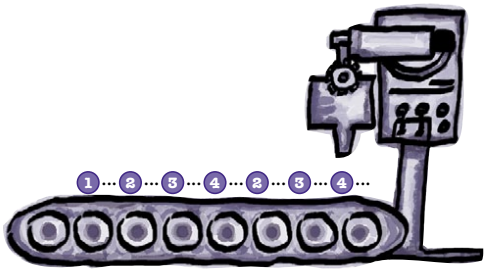
Ryan Benedetti / Ronan Cranley, jQuery von Kopf bis Fuß, O´Reilly, ISBN 9783868991895

jQuery und JavaScript

6 ~~Luke~~ jQuery, Ich bin dein Vater!

Alles kann jQuery nicht alleine tun. Auch wenn jQuery eine JavaScript-Bibliothek ist, kann es nicht alles, was seine Ursprungssprache kann. In diesem Kapitel werden wir uns mit einigen JavaScript-Fähigkeiten befassen, die für die Erstellung fesselnder Websites gebraucht werden. Wir werden sehen, wie jQuery diese Merkmale nutzen kann, um eigene Listen und Objekte zu erstellen und darüber Schleifen auszuführen, wodurch Ihr Leben ein gutes Stück leichter wird.

Frischer Wind für die Head First Lounge	216
In Objekten ist das Speichern noch schlauer gelöst	218
Eigene Objekte bauen	219
Mit Konstruktoren wiederverwendbare Objekte erstellen	220
Objekte benutzen	221
Die Seite ersetllen	222
Die Rückkehr der Arrays	225
Auf Arrays zugreifen	226
Arrayelemente aktualisieren und hinzufügen	227
Eine Aktion nochmal durchführen (und nochmal und nochmal …)	229
Die Nadel im Heuhaufen finden	232
Zeit für Entscheidungen … wieder einmal!	239
Vergleichsoperatoren und logische Operatoren	240
jQuery zum Aufräumen benutzen …	246
Etwas mehr Nervenkitzel, bitte …	250
Ihr jQuery-/JavaScript-Werkzeugkasten	252



7 Eigene Funktionen für Ihre Effekte

Tu doch mal was für mich!

Kombinieren Sie jQuery-Effekte mit JavaScript-Funktionen, um Ihren Code – und damit Ihre Web-Applikation – effizienter, effektiver und noch *leistungsfähiger* zu machen. In diesem Kapitel werden Sie sehen, wie Sie Ihre jQuery-Effekte durch die Verwendung von **Browser Events** und **zeitgesteuerten Funktionen**, sowie die **Organisation und Wiederverwendbarkeit** Ihrer selbst erstellten JavaScript-Funktionen verbessern können.

Ein Sturm braut sich zusammen	254
Wir haben eine Monster-Funktion geschaffen …	255
Mit dem window-Objekt die Kontrolle über zeitgesteuerte Effekte übernehmen	256
Mit onblur und onfocus auf Browser-Events reagieren	259
Timer-Methoden sagen Ihren Funktionen, wann sie ausgeführt werden sollen	263
Schreiben Sie die Funktionen blitze_anhalten und blitze_starten	266
Neue Funktionen für die Monstermacher	274
Noch mehr Zufall	275
Sie kennen die aktuelle Position schon ...	276
… und auch die Funktion zufallszahl_erzeugen	276
Dinge relativ zur aktuellen Position bewegen	280
Die Monstermacher, Version 2 ist ein Hit!	289
Ihr jQuery-Werkzeugkasten	290

Ryan Benedetti / Ronan Cranley, jQuery von Kopf bis Fuß, O´Reilly, ISBN 9783868991895

jQuery und Ajax

8 Gib' doch mal die Daten rüber

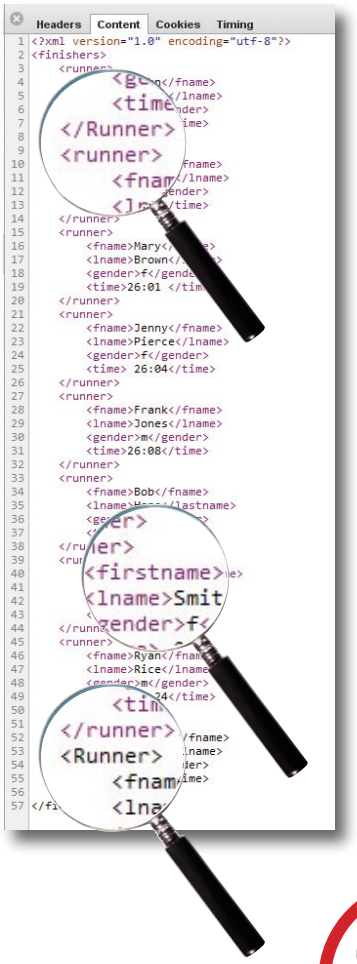
CSS- und DOM-Tricks mit jQuery machen schon Spaß, aber irgendwann müssen Sie Informationen (oder Daten) von einem Server abfragen und darstellen. Möglicherweise wollen Sie sogar bestimmte Teile einer Seite mit den Informationen vom Server aktualisieren, ohne hierfür die ganze Seite neu laden zu müssen. Und da kommt Ajax ins Spiel. In Kombination mit jQuery und JavaScript sind auch diese Dinge möglich. In diesem Kapitel lernen Sie, wie Sie mit jQuery Ajax-Abfragen an den Server schicken können und was Sie mit den zurückgegebenen Antworten anfangen können.

Bringen Sie das Bit to Byte-Rennen ins neue Jahrhundert	292
Ein Blick auf die Seite des letzten Jahres	293
Jetzt wird's dynamisch	296
Das ALTE Web trifft das NEUE Web	297
Ajax verstehen	298
Was ist Ajax?	298
Der X-Faktor	299
Daten mit der ajax-Methode abfragen	304
XML-Daten parsen	306
Funktionen zeitgesteuert aufrufen	310
Selbstreferenzierende Funktionen	311
Mehr vom Server abfragen	314
Wie spät ist es?	315
Zeitgesteuerte Events auf der Seite abschalten	320
Ihr jQuery-/Ajax-Werkzeugkasten	324

handling JSON data

9 Darf ich vorstellen: Client – Server

Das Lesen von Daten aus einer XML-Datei ist zwar sehr nützlich, immer bringt's das aber auch nicht. Mit einem effizienteren Format (JavaScript Object Notation, auch bekannt als JSON), kann der Datenaustausch mit dem Server deutlich erleichtert werden. Außerdem lässt JSON nicht nur leichter erzeugen, sondern auch lesen. Mit Hilfe von jQuery, PHP und SQL werden Sie lernen, eine Datenbank anzulegen, um darin Informationen zu speichern, die Sie später mit JSON auslesen und mit Hilfe von jQuery auf dem Bildschirm darstellen können. Für Web-Applikationen ist das eine echte Supermacht.



Die MegaCorp-Marketingabteilung von Webville kennt kein XML	326
XML-Fehler bringen die Seite zum Absturz	327
Die Daten über eine Webseite eingeben	328
Was tun mit den Daten?	331
Daten vor dem Versand formatieren	332
Daten an den Server schicken	333
Daten in einer MySQL-Datenbank speichern	335
Eine Datenbank zum Speichern der Läuferinformationen verwenden	336
Anatomie einer INSERT-Anweisung	338
Mit PHP auf die Daten zugreifen	341
Auf dem Server mit POST-Daten umgehen	342
Datenbankverbindungen mit PHP	343
Daten per SELECT aus der Datenbank abfragen	345
Daten mit PHP abfragen	347
JSON hilft	350
jQuery + JSON = Großartig	351
Ein paar PHP-Regeln ...	352
(Noch) ein paar PHP-Regeln …	353
Ausgaben mit PHP formatieren	354
Auf Daten im JSON-Objekt zugreifen	361
Datensäuberung und -validierung in PHP	364
Ihr jQuery-/Ajax-/PHP-/MySQL-Werkzeugkasten	369

Ryan Benedetti / Ronan Cranley, jQuery von Kopf bis Fuß, O´Reilly, ISBN 9783868991895

jQuery UI

10 Formulare aufwerten

Das Web lebt und stirbt mit den Daten seiner Benutzer. Das Einsammeln von Nutzerdaten ist keine Kleinigkeit und kann für Webentwickler ziemlich zeitaufwändig werden. Wie jQuery die Effizienz von Ajax-, PHP- und MySQL-Webapplikationen steigern kann, haben wir schon gesehen. In diesem Kapitel geht es um die Verwendung von jQuery für die Optimierung von Formularelementen für die Dateneingabe. Und wenn wir schon dabei sind, gibt es auch gleich noch eine gesunde Menge jQuery UI, der offiziellen UI-Bibliothek für jQuery mit dazu.

Cryptozoologists.org muss neu gestaltet werden	372
Pimpen Sie Ihr HTML-Formular	373
Durch jQuery UI Zeit und Kopfschmerzen beim Programmieren sparen	376
Was ist im jQuery UI-Paket enthalten?	380
Ein Kalender-Widget zur Eingabe des Sichtungsdatums erstellen	381
jQuery UI hinter den Kulissen	382
Widgets können über Optionen angepasst werden	383
Die Radiobuttons ansprechender gestalten	386
Nummerische Werte über einen Schieberegler eingeben	390
Computer mischen Farben aus Rot, Grün und Blau	399
Die palette_aktualisieren-Funktion erstellen	402
Und noch etwas …	406
Ihr jQuery-Werkzeugkasten	410

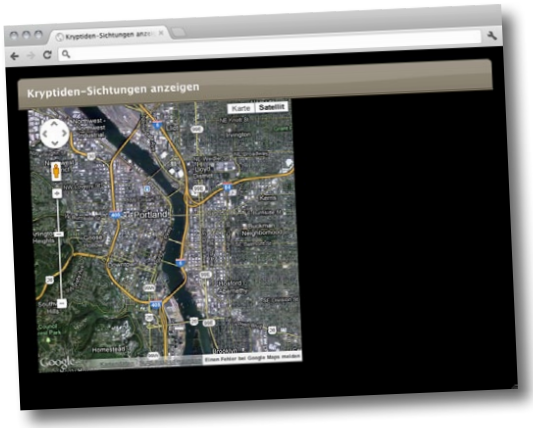
jQuery und APIs

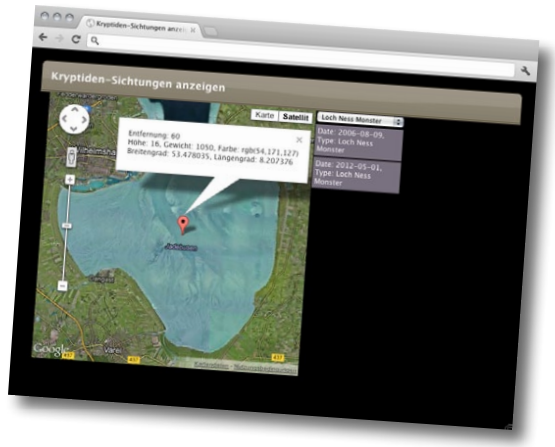
11 Objekte, nichts als Objekte!

Selbst ein talentierter Entwickler wie Sie kann nicht alles alleine machen …

Wir haben gesehen, wie wir jQuery-Plugins wie jQuery UI oder die Tab-Navigation in unsere Seiten einbinden können, um Web-Applikationen zu verbessern. Die nächste Entwicklungsstufe besteht darin, Werkzeuge aus dem Internet und Informationen von Größen wie Google, Twitter oder Yahoo! zu verwenden. Hierfür brauchen wir aber etwas … mehr. Diese Firmen stellen ihre Dienste über APIs (application programming interfaces, Programmierschnittstellen) zur Verfügung, die Sie auf Ihrer Seite nutzen können. In diesem Kapitel werden wir uns mit einigen Grundlagen der Arbeit mit APIs befassen. Als Beispiel benutzen wir die beliebte Google Maps API.

Wo ist Nessie?	412
Die Google Maps API	414
APIs verwenden Objekte	415
Eine Google Maps-Karte in Ihre Seite einbinden	417
JSON-Daten mit SQL und PHP erzeugen	420
Punkte auf der Karte markieren	424
Checkliste für mehrere Kreaturen	428
Karten-Events abfangen	438
Sie haben es geschafft!	442
Ihr jQuery API-Werkzeugkasten	445





Ryan Benedetti / Ronan Cranley, jQuery von Kopf bis Fuß, O´Reilly, ISBN 9783868991895

Übrig gebliebenes

Die 10 wichtigsten Dinge (die wir nicht behandelt haben)

Trotz der vielen Dinge in diesem Buch konnten wir nicht alles behandeln. Es gibt noch jede Menge weiterer guter Dinge in jQuery und JavaScript, die einfach nicht mehr in dieses Buch gepasst haben. Wir möchten Ihnen hier zumindest davon erzählen, um Sie auf die anderen Facetten von jQuery vorzubereiten, die Ihnen unterwegs begegnen können.

#1. Jede Einzelheit der jQuery-Bibliothek 448
#2. jQuery CDNs 451
#3. Der jQuery-Namensraum: die noConflict-Methode 452
#4. jQuery-Code debuggen 453
#5. Fortgeschrittene Animationen: Warteschlangen 454
#6. Formulareingaben auf Gültigkeit überprüfen 455
#7. jQuery UI-Effekte 456
#8. Eigene jQuery-Plugins schreiben 457
#9. Fortgeschrittenes JavaScript: Closures 458
#10. Templates 459

Eine Entwicklungsumgebung einrichten

Bereit für große Zeiten

Sie brauchen einen Ort, an dem Sie Ihre neuen PHP-Fähigkeiten ausprobieren können, ohne Ihre Daten im Web preiszugeben. Es ist prinzipiell eine gute Idee, Ihre PHP-Applikation an einem sicheren Ort zu entwickeln, bevor sie auf die (WW)Welt losgelassen wird. In diesem Anhang finden Sie eine Anleitung zur Installation eines Webservers sowie von MySQL und PHP, damit Sie einen sicheren Ort zum Üben und Arbeiten haben.

Eine PHP-Entwicklungsumgebung einrichten	462
Herausfinden, was schon installiert ist	462
Haben Sie einen Webserver?	463
Ist PHP installiert? Welche Version?	463
Ist MySQL installiert? Welche Version?	464
Den Webserver starten	465
Apache-Installation … Fortsetzung	466
PHP-Installation	466
PHP-Installationsschritte	467
PHP-Installationsschritte … Fortsetzung	468
MySQL installieren	468
Arbeitsschritte zur Installation von MySQL unter Windows	469
PHP unter Mac OS X einrichten	474
Arbeitsschritte zur Installation von MySQL unter Mac OS X	474

Ryan Benedetti / Ronan Cranley, jQuery von Kopf bis Fuß, O´Reilly, ISBN 9783868991895

4 Webseiten mit jQuery verändern

Mod the DOM

Dass die Seite fertig geladen ist, heißt nicht, dass sie deshalb die gleiche Struktur behalten muss. In Kapitel 1 haben wir bereits gesehen, wie beim Laden einer Seite das DOM aufgebaut wird, um die Struktur der Seite zu erstellen. In diesem Kapitel wollen wir sehen, wie man sich in der DOM-Struktur auf- und abbewegt. Wir werden Ihnen zeigen, wie man mit der Hierarchie der Elemente und mit Eltern-Kind-Beziehungen arbeitet, um die Seitenstruktur nach Bedarf mit jQuery zu verändern.

Dies ist ein Auszug aus dem Buch "jQuery von Kopf bis Fuß", ISBN 978-3-86899-189-5
http://www.oreilly.de/catalog/hfjqueryger/
Dieser Auszug unterliegt dem Urheberrecht. © O'Reilly Verlag 2011

Das Webville-Restaurant braucht eine interaktive Speisekarte

Alexandra ist die Chefköchin des Webville-Restaurants. Sie hat einen Auftrag für Sie. Bisher hat sie mehrere Webseiten für verschiedene Versionen ihrer Speisekarte gepflegt: eine normale und eine vegetarische Variante. Jetzt hätte sie gerne eine Karte, die das Menü bei Bedarf an die Wünsche der vegetarischen Gäste anpasst.

Dies ist ein Auszug aus dem Buch "jQuery von Kopf bis Fuß", ISBN 978-3-86899-189-5
http://www.oreilly.de/catalog/hfjqueryger/
Dieser Auszug unterliegt dem Urheberrecht. © O'Reilly Verlag 2011

Ryan Benedetti / Ronan Cranley, jQuery von Kopf bis Fuß, O´Reilly, ISBN 9783868991895

Die vegetarische Version

Alexandra möchte, dass Sie die folgenden Dinge für sie erledigen:

Wir hätten gerne einen »Für Vegetarier«-Button. Wenn er angeklickt wird, sollen die fleischlichen Bestandteile in der Web-Speisekarte automatisch gegen vegetarische Entsprechungen ausgetauscht werden.

Die Ersetzungen sollen so funktionieren:

- Für Fischgerichte gibt es keinen vegetarischen Ersatz. Die müssen also entfernt werden.
- Als vegetarische Alternative für Hamburger bieten wir Portobello-Zuchtchampignons an.
- Für alle unsere Fleisch- und Eiergerichte außer Hackfleischgerichten bieten wir Tofu als vegetarische Alternative an.
- Wir benötigen einen Button, der das Menü wieder in seinen Originalzustand versetzt.

P.S. Es wäre schön, wenn außerdem ein Blatt-Symbol neben den vegetarischen Alternativgerichten angezeigt werden könnte.

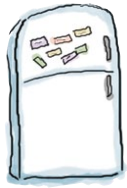
Ich habe den Webdesigner dazu veranlasst, Dir die Dateien für das aktuelle Menü zu schicken, damit Du schon mal anfangen kannst.

Webville-Restaurant

Diesmal gibt es hier keine Übung, auch wenn Sie das vielleicht erwartet haben. Schreiben Sie die Anforderungsliste aber auf jeden Fall noch einmal in eigenen Worten auf, damit Sie wissen, was hier passieren soll.

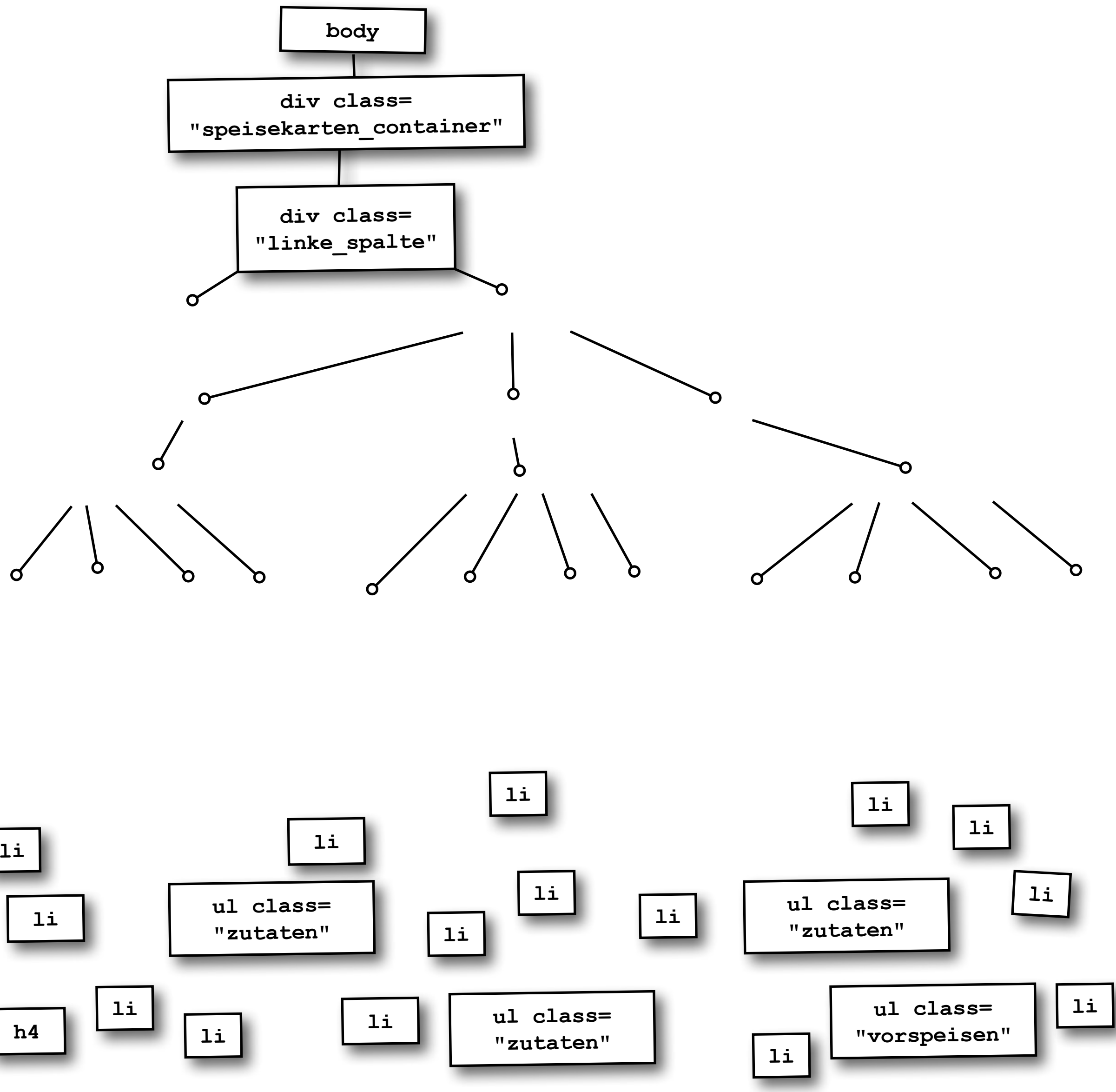
Bevor wir anfangen, jQuery-Code zu schreiben, wollen wir zunächst einen Blick auf die HTML- und CSS-Dateien werfen, die uns der Webdesigner geschickt hat. Dann können wir feststellen, ob Stil und Struktur für unsere Aufgabe verwendbar sind.

Dies ist ein Auszug aus dem Buch "jQuery von Kopf bis Fuß", ISBN 978-3-86899-189-5
http://www.oreilly.de/catalog/hfjqueryger/
Dieser Auszug unterliegt dem Urheberrecht. © O'Reilly Verlag 2011



DOM-Baum-Magneten

Machen Sie sich ein Bild von der momentanen Struktur der Speisekarte, indem Sie ein Diagramm des DOM-Baumes erstellen. Unten finden Sie die Magneten für die einzelnen Elemente, die zum Vervollständigen des Baumes gebraucht werden. Überall, wo Sie im Diagramm einen Kreis sehen, gehört ein Magnet hin. Ein paar haben wir schon für Sie angebracht.



Dies ist ein Auszug aus dem Buch "jQuery von Kopf bis Fuß", ISBN 978-3-86899-189-5
http://www.oreilly.de/catalog/hfjqueryger/
Dieser Auszug unterliegt dem Urheberrecht. © O'Reilly Verlag 2011

Ryan Benedetti / Ronan Cranley, jQuery von Kopf bis Fuß, O´Reilly, ISBN 9783868991895

Das hier ist nur ein Ausschnitt aus der gesamten HTML-Seite,

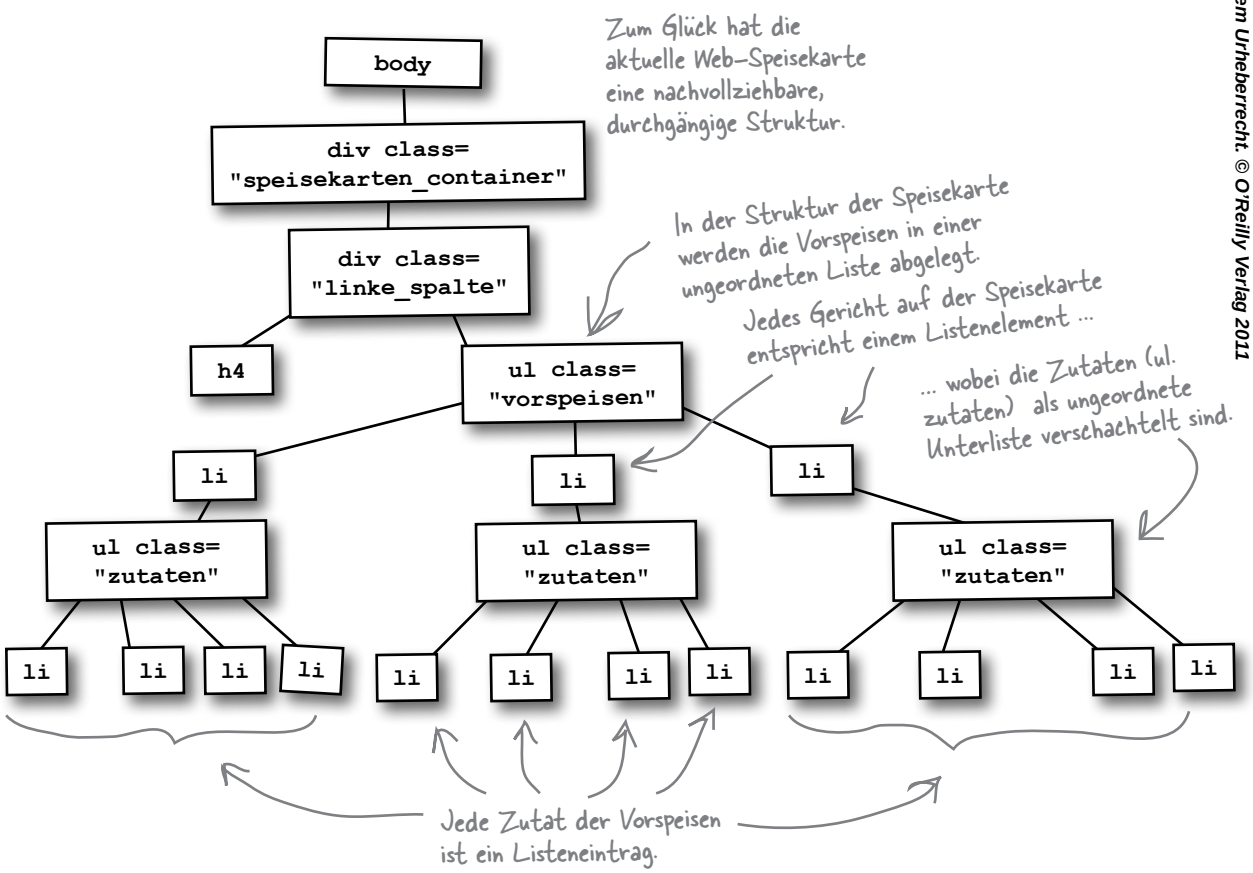
```
<body>
 <div id="speisekarten_container">
  <div class="linke_spalte">
   <h4>Vorspeisen</h4>
   <ul class="vorspeisen">
       <li>Heilbutt (Thai-Style)
              <ul class="zutaten">
                     <li>Kokosmilch</li>
                     <li>Gebratener Heilbutt</li>
                     <li>Herbstgemüse</li>
                     <li>Thai-Gewürze</li>
              </ul>
       </li>
       <li>Panini vom Grill
              <ul class="zutaten">
                     <li>Prosciutto</li>
                     <li>Provolone (Hartkäse)</li>
                     <li>Avocado</li>
                     <li>Sauerteig-Brötchen</li>
              </ul>
       </li>
       <li>Southwest Slider
              <ul class="zutaten">
                     <li>Ganze Chillis</li>
                     <li>Hamburger</li>
                     <li>Pepperjack-Käse (mit Pfeffer)</li>
                     <li>Mehrkorn-Brötchen</li>
              </ul>
       </li>
   </ul>
  </div>
 </div>
</body>
```

index.html

Dies ist ein Auszug aus dem Buch "jQuery von Kopf bis Fuß", ISBN 978-3-86899-189-5
http://www.oreilly.de/catalog/hfjqueryger/
Dieser Auszug unterliegt dem Urheberrecht. © O'Reilly Verlag 2011

DOM-Baum-Magneten: Lösung

Es sieht so aus, als seien die Zutaten als Kindelemente in Form von Listeneinträgen der übergeordneten Liste für die Vorspeisenliste angelegt. Sie sind nicht besonders deutlich oder gar einmalig markiert, oder?



Dies ist ein Auszug aus dem Buch "jQuery von Kopf bis Fuß", ISBN 978-3-86899-189-5
http://www.oreilly.de/catalog/hfjqueryger/
Dieser Auszug unterliegt dem Urheberrecht. © O'Reilly Verlag 2011

Ryan Benedetti / Ronan Cranley, jQuery von Kopf bis Fuß, O´Reilly, ISBN 9783868991895

```
<body>
 <div id="speisekarten_container">
  <div class="linke_spalte">
   <h4>Vorspeisen</h4>
   <ul class="vorspeisen">
       <li>Heilbutt (Thai-Style)
              <ul class="zutaten">
                     <li>Kokosmilch</li>
                     <li>Gebratener Heilbutt</li>
                     <li>Brühe mit Zitronengras</li>
                     <li>Thai-Gewürze</li>
              </ul>
       </li>
       <li>Panini vom Grill
              <ul class="zutaten">
                     <li>Prosciutto</li>
                     <li>Provolone (Hartkäse)</li>
                     <li>Avocado</li>
                     <li>Sauerteig-Brötchen</li>
              </ul>
       </li>
       <li>Southwest Slider
              <ul class="zutaten">
                     <li>Ganze Chillis</li>
                     <li>Hamburger</li>
                     <li>Pepperjack-Käse (mit Pfeffer)</li>
                     <li>Mehrkorn-Brötchen</li>
              </ul>
       </li>
   </ul>
  </div>
 </div>
</body>
```

Das hier ist nur ein Ausschnitt aus der gesamten HTML-Seite.

index.html

KOPF-NUSS

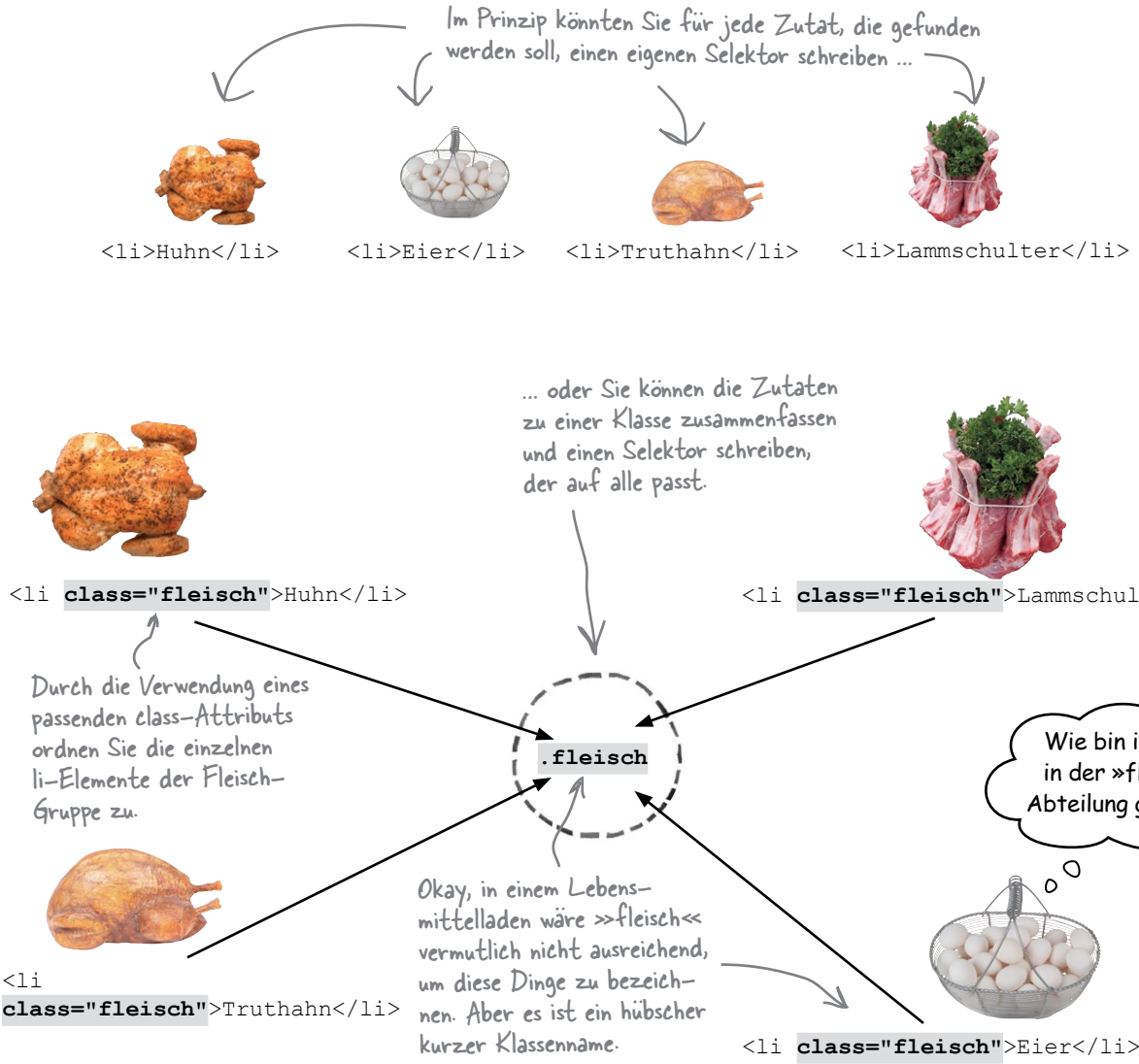
Eine reguläre Webseitenstruktur (HTML) erleichtert das Schreiben von jQuery-Code. Allerdings sind die Zutaten, die wir finden wollen, nicht speziell markiert. Was können wir tun, damit die Elemente leichter ausgewählt werden können?

Dies ist ein Auszug aus dem Buch "jQuery von Kopf bis Fuß", ISBN 978-3-86899-189-5
http://www.oreilly.de/catalog/hfjqueryger/
Dieser Auszug unterliegt dem Urheberrecht. © O'Reilly Verlag 2011

Elemente mit Klasse

Wie Sie aus den vorigen Kapiteln bereits wissen, können gut geschriebener HTML- und CSS-Code dabei helfen, dass jQuery die Elemente in der Webseite effizienter findet. Damit die Struktur so richtig gut wird, sollten wir das Stylesheet mit passenden Klassen und IDs versehen. Im HTML-Code fügen wir dann die passenden id- und class-Attribute ein. Dadurch wird die Auswahl der Elemente wesentlich einfacher und Sie brauchen später nicht so viel Zeit zum Programmieren.

In jQuery sind die Selektoren nicht nur dafür da, das Aussehen von Seiten zu steuern. Mit Selektoren kann jQuery auch bestimmte Elemente in der Seite finden.



Ryan Benedetti / Ronan Cranley, jQuery von Kopf bis Fuß, O´Reilly, ISBN 9783868991895

Dies ist ein Auszug aus dem Buch "jQuery von Kopf bis Fuß", ISBN 978-3-86899-189-5
http://www.oreilly.de/catalog/hfjqueryger/
Dieser Auszug unterliegt dem Urheberrecht. © O'Reilly Verlag 2011

Finden Sie die Zutaten in der Speisekarte, für die die Chefköchin eine Alternative anbieten möchte, und versehen Sie sie mit der passenden Klasse (fisch, fleisch oder hamburger). Braucht eine Zutat keine Klasse, lassen Sie die Zeile frei. Der HTML-Code ist hier so dargestellt, wie er nachher auch auf dem Bildschirm erscheint.

```
<li>Heilbutt (Thai-Style)
  <ul class="zutaten">
    <li.....>Kokosmilch</li>
    <li.....>Gebratener Heilbutt</li>
    <li.....>Brühe mit Zitronengras</li>
    <li.....>Gemüse</li>
    <li.....>Thai-Gewürze</li>
  </ul>
</li>

<li>Geschmorte Köstlichkeiten
  <ul class="zutaten">
    <li.....>Lammschulter</li>
    <li.....>Zwiebeln</li>
    <li.....>Karotten</li>
    <li.....>Junge Rüben</li>
    <li.....>Bratensaft</li>
  </ul>
</li>

<li>Panini vom Grill
  <ul class="zutaten">
    <li.....>Prosciutto</li>
    <li.....>Provolone (Hartkäse)</li>
    <li.....>Avocado</li>
    <li.....>Kirschtomaten</li>
    <li.....>Sauerteig-Brötchen</li>
    <li.....>Shoestring Fries </li>
  </ul>
</li>

<li>Unsere Miniburger
  <ul class="zutaten">
    <li.....>Aubergine</li>
    <li.....>Zucchini</li>
    <li.....>Hamburger</li>
    <li.....>Balsamessig</li>
    <li.....>Zwiebeln</li>
    <li.....>Karotten</li>
    <li.....>Mehrkorn-Brötchen</li>
    <li.....>goat cheese</li>
  </ul>
</li>

<li>Frittata (Ital. Omelett)
  <ul class="zutaten">
    <li.....>Eier</li>
    <li.....>Asiagokäse</li>
    <li.....>Kartoffeln </li>
  </ul>
</li>

<li>Kokossuppe
  <ul class="zutaten">
    <li.....>Kokosmilch</li>
    <li.....>Huhn</li>
    <li.....>Gemüsebrühe</li>
  </ul>
</li>

<li>Soup Du Jour
  <ul class="zutaten">
    <li.....>Steak vom Grill</li>
    <li.....>Champignons</li>
    <li.....>Gemüse</li>
    <li.....>Gemüsebrühe </li>
  </ul>
</li>

<li>Suppe scharf und sauer
  <ul class="zutaten">
    <li class="fleisch">Geröstetes Schwein</li>
    <li.....>Karotten</li>
    <li.....>Chinesische Pilze</li>
    <li.....>Chilli</li>
    <li.....>Gemüsebrühe </li>
  </ul>
</li>

<li>Avocadobrötchen
    <ul class="zutaten">
    <li.....>Avocado</li>
    <li.....>Ganze Chillis</li>
    <li.....>Süße rote Paprika</li>
    <li.....>Ingwersauce</li>
  </ul>
</li>
```

Dies ist ein Auszug aus dem Buch "jQuery von Kopf bis Fuß", ISBN 978-3-86899-189-5
http://www.oreilly.de/catalog/hfjqueryger/
Dieser Auszug unterliegt dem Urheberrecht. © O'Reilly Verlag 2011

LÖSUNG ZUR ÜBUNG

Finden Sie die Zutaten in der Speisekarte, für die die Chefköchin eine Alternative anbieten möchte, und versehen Sie sie mit der passenden Klasse (fisch, fleisch oder hamburger). Braucht eine Zutat keine Klasse, lassen Sie die Zeile frei.

```
<li>Heilbutt (Thai-Style)
  <ul class="zutaten">
    <li                    >Kokosmilch</li>
    <li class="fisch"      >Gebratener Heilbutt</li>
    <li                    >Brühe mit Zitronengras</li>
    <li                    >Gemüse</li>
    <li                    >Thai-Gewürze</li>
  </ul>
</li>

<li>Geschmorte Köstlichkeiten
  <ul class="zutaten">
    <li class="fleisch"    >Lammschulter</li>
    <li                    >Zwiebeln</li>
    <li                    >Karotten</li>
    <li                    >Junge Rüben</li>
    <li                    >Bratensaft</li>
  </ul>
</li>

<li>Panini vom Grill
  <ul class="zutaten">
    <li class="fleisch"    >Prosciutto</li>
    <li                    >Provolone (Hartkäse)</li>
    <li                    >Avocado</li>
    <li                    >Kirschtomaten</li>
    <li                    >Sauerteig-Brötchen</li>
    <li                    >Shoestring Fries </li>
  </ul>
</li>

<li>Unsere Miniburger
  <ul class="zutaten">
    <li                    >Aubergine</li>
    <li                    >Zucchini</li>
    <li class="hamburger"  >Hamburger</li>
    <li                    >Balsamessig</li>
    <li                    >Zwiebeln</li>
    <li                    >Karotten</li>
    <li                    >Mehrkorn-Brötchen</li>
    <li                    >goat cheese</li>
  </ul>
</li>

<li>Frittata (Ital. Omelett)
  <ul class="zutaten">
    <li class="fleisch"    >Eier</li>
    <li                    >Asiagokäse</li>
    <li                    >Kartoffeln </li>
  </ul>
</li>

<li>Kokossuppe
  <ul class="zutaten">
    <li                    >Kokosmilch</li>
    <li class="fleisch"    >Huhn</li>
    <li                    >Gemüsebrühe</li>
  </ul>
</li>

<li>Soup Du Jour
  <ul class="zutaten">
    <li class="fleisch"    >Steak vom Grill</li>
    <li                    >Champignons</li>
    <li                    >Gemüse</li>
    <li                    >Gemüsebrühe </li>
  </ul>
</li>

<li>Suppe scharf und sauer
  <ul class="zutaten">
    <li class="fleisch"    >Geröstetes Schwein</li>
    <li                    >Karotten</li>
    <li                    >Chinesische Pilze</li>
    <li                    >Chilli</li>
    <li                    >Gemüsebrühe </li>
  </ul>
</li>

<li>Avocadobrötchen
    <ul class="zutaten">
    <li                    >Avocado</li>
    <li                    >Ganze Chillis</li>
    <li                    >Süße rote Paprika</li>
    <li                    >Ingwersauce</li>
  </ul>
</li>
```

Dies ist ein Auszug aus dem Buch "jQuery von Kopf bis Fuß", ISBN 978-3-86899-189-5
http://www.oreilly.de/catalog/hfjqueryger/
Dieser Auszug unterliegt dem Urheberrecht. © O'Reilly Verlag 2011

Ryan Benedetti / Ronan Cranley, jQuery von Kopf bis Fuß, O´Reilly, ISBN 9783868991895

Und jetzt an die Umsetzung

Jetzt, wo wir so weit alles vorbereitet haben, wollen wir uns die Serviette mit den Anforderungen der Chefköchin noch einmal ansehen. Der nächste Schritt besteht in der Erstellung von zwei Buttons.

– Wir hätten gerne einen »Für Vegetarier«-Button. Wenn er angeklickt wird, sollen die fleischlichen Bestandteile in der Web-Speisekarte automatisch gegen vegetarische Entsprechungen ausgetauscht werden.

– Wir benötigen einen Button, der das Menü wieder in seinen Originalzustand versetzt.

Spitzen Sie Ihren Bleistift

Aktualisieren Sie Seitenstruktur und Skript, um die beiden gewünschten Buttons einzubauen. Geben Sie dem »Für Vegetarier«-Button die ID `vegetarisch_an` und dem »Zurücksetzen«-Button die ID `zuruecksetzen`.

```
<div class="alternativen_waehlen">
  <h2>Unser Menü</h2>
    <ul>
      <li class="nav">.....</li>
      <li class="nav">.....</li>
    </ul>
</div>
```

index.html

```
$(document).ready(function() {
    var v = false;

    .....
        if (v == false){

          v = true}
    });//ende button

    .....
        if (v == true){

          v = false;}
    });//ende button
});//ende document ready
```

meine_skripts.js

Dies ist ein Auszug aus dem Buch "jQuery von Kopf bis Fuß", ISBN 978-3-86899-189-5
http://www.oreilly.de/catalog/hfjqueryger/
Dieser Auszug unterliegt dem Urheberrecht © O'Reilly Verlag 2011

Spitzen Sie Ihren Bleistift
Lösung

Aktualisieren Sie Seitenstruktur und Skript, um die beiden gewünschten Buttons einzubauen. Geben Sie dem »Für Vegetarier«-Button die ID `vegetarisch_an` und dem »Zurücksetzen«-Button die ID `zuruecksetzen`.

```
<div class="alternativen_waehlen">
  <h2>Unser Menü</h2>
    <ul>
      <li class="nav"> <button id="vegetarisch_an">Go Vegetarian</button> </li>
      <li class="nav"> <button id="zuruecksetzen">Restore Menu</button> </li>
    </ul>
</div>
```

Button-Elemente mit den IDs vegetarisch_an und zuruecksetzen erzeugen.

index.html

Ein etwas spezifischerer Selektor, der den Elementtyp und dessen ID benutzt

```
$(document).ready(function() {
    var v = f;
    $("button#vegetarisch_an").click(function(){
        if (v == false){
          v = true;}
    });//ende button
    $("button#zuruecksetzen").click(function(){
        if (v == true){
          v = false;}
    });//ende button
});//ende document ready
```

Die Buttons mit der click-Methode versehen.

meine_skripts.js

Dies ist ein Auszug aus dem Buch "jQuery von Kopf bis Fuß", ISBN 978-3-86899-189-5
http://www.oreilly.de/catalog/hfjqueryger/
Dieser Auszug unterliegt dem Urheberrecht. © O'Reilly Verlag 2011

Ryan Benedetti / Ronan Cranley, jQuery von Kopf bis Fuß, O´Reilly, ISBN 9783868991895

Was kommt jetzt?

Das ging schnell. Die zwei Buttons sind fertig. Diese Dinge können können wir von der Serviette streichen und uns mit den Dingen beschäftigen, die der »Für Vegetarier«-Button tun soll.

- Wir hätten gerne einen »Für Vegetarier«-Button. Wenn er angeklickt wird, sollen die fleischlichen Bestandteile in der Web-Speisekarte automatisch gegen vegetarische Entsprechungen ausgetauscht werden.

- Wir benötigen einen Button, der das Menü wieder in seinen Originalzustand versetzt.

Die Ersetzungen sollen so funktionieren:

- Für Fischgerichte gibt es keinen vegetarischen Ersatz. Die müssen also entfernt werden.

- Als vegetarische Alternative für Hamburger bieten wir Portobello-Zuchtchampignons an.

- Für alle unsere Fleisch- und Eiergerichte außer Hackfleischgerichten bieten wir Tofu als vegetarische Alternative an.

ÜBUNG

Schreiben Sie in eigenen Worten auf, was der »Für Vegetarier«-Button tun soll.

1. .....
   .....

2. .....
   .....

3. .....
   .....

Dies ist ein Auszug aus dem Buch "jQuery von Kopf bis Fuß", ISBN 978-3-86899-189-5
http://www.oreilly.de/catalog/hfjqueryger/
Dieser Auszug unterliegt dem Urheberrecht. © O'Reilly Verlag 2011

LÖSUNG ZUR ÜBUNG

Schreiben Sie in eigenen Worten auf, was der »Für Vegetarier«-Button tun soll.

1. Finde Elemente mit der Klasse >>fisch<< und entferne sie aus der Liste.
2. Finde li-Elemente mit der Klasse >>hamburger<< und ersetze sie durch Portobello-Champignons.
3. Finde li-Elemente mit der Klasse >>fleisch<< und ersetze sie durch Tofu.

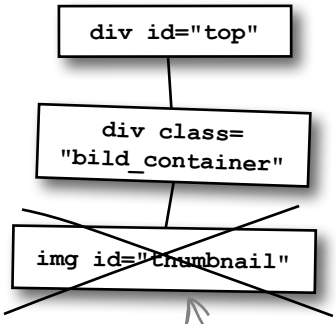
Machen Sie sich keine Sorgen, wenn Ihre Antworten nicht ganz die gleichen wie unsere sind. Die Übersetzung der Anforderungen in Code, die für die App benötigt werden, braucht Übung.

Dies ist ein Auszug aus dem Buch "jQuery von Kopf bis Fuß", ISBN 978-3-86899-189-5
http://www.oreilly.de/catalog/jqueryger/
Dieser Auszug unterliegt dem Urheberrecht. © O'Reilly Verlag 2011

Unsere nächste Aufgabe besteht darin, den ersten Punkt oben abzuarbeiten: Listeneinträge mit der Klasse »fisch« zu finden und aus der Speisekarte zu entfernen. In Kapitel 2 haben wir Elemente mit Klassenselektoren ausgewählt und die `remove`-Methode benutzt, um sie aus dem DOM zu entfernen.

In jQuery gibt es zusätzlich noch die `detach`-Methode. Beide Methoden können Elemente aus dem DOM entfernen. Worin besteht der Unterschied?

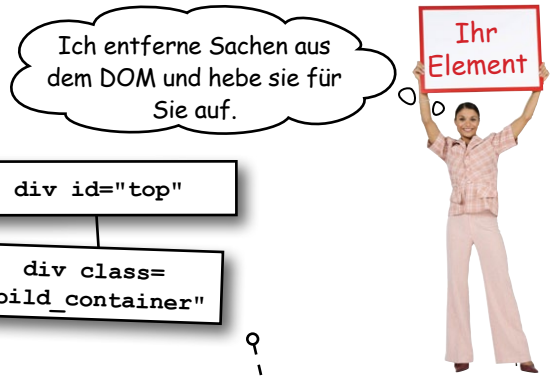
remove

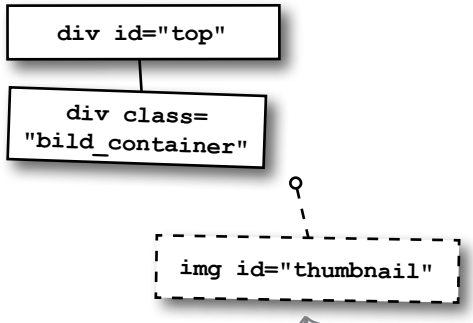


Die remove-Methode entfernt ein Element aus dem DOM.

```
$("img#thumbnail").remove();
```

detach





Die detach-Methode enternt das gewählte Element aus dem DOM und bewahrt es auf, damit es später wieder eingefügt werden kann.

```
$("img#thumbnail").detach();
```

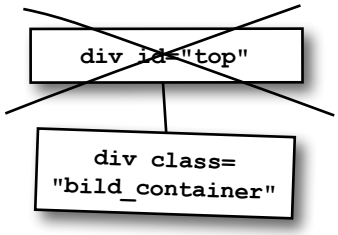

Ryan Benedetti / Ronan Cranley, jQuery von Kopf bis Fuß, O´Reilly, ISBN 9783868991895

Spitzen Sie Ihren Bleistift

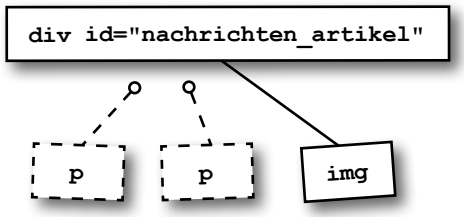
Benutzen Sie den freien Platz, um den Selektor sowie die `remove`- oder `detach`-Methode aufzuschreiben, die das Ergebnis auf der linken Seite hervorbringen.

DOM-Ergebnis

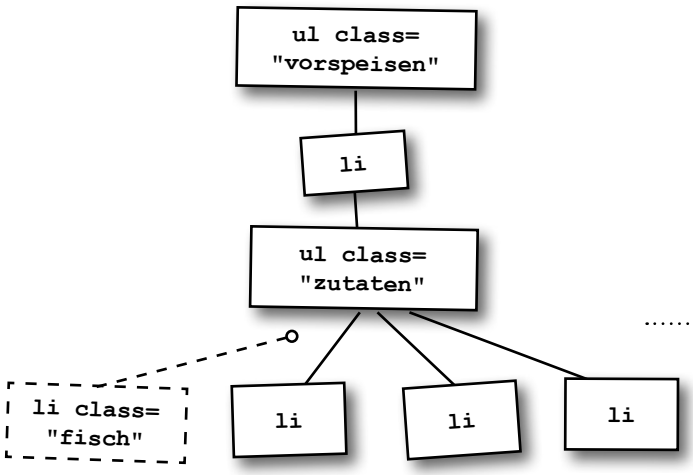
jQuery-Anweisung



.....



.....



.....

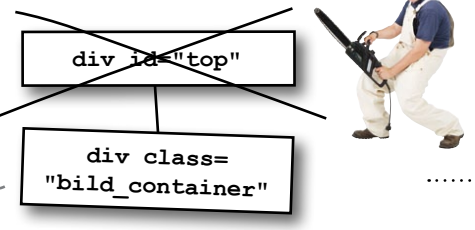
Dies ist ein Auszug aus dem Buch "jQuery von Kopf bis Fuß", ISBN 978-3-86899-189-5
http://www.oreilly.de/catalog/hfjquerygerger/
Dieser Auszug unterliegt dem Urheberrecht. © O'Reilly Verlag 2011

Spitzen Sie Ihren Bleistift
Lösung

Benutzen Sie den freien Platz, um den Selektor sowie die `remove`- oder `detach`-Methode aufzuschreiben, die das Ergebnis auf der linken Seite hervorbringen.

Wenn Sie remove oder detach auf ein Element anwenden, werden auch dessen Nachfahren-Elemente entfernt.

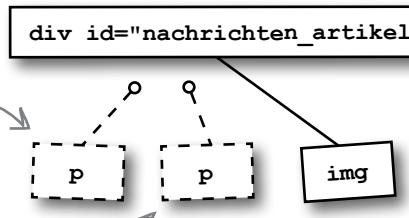
DOM-Ergebnis



jQuery-Anweisung

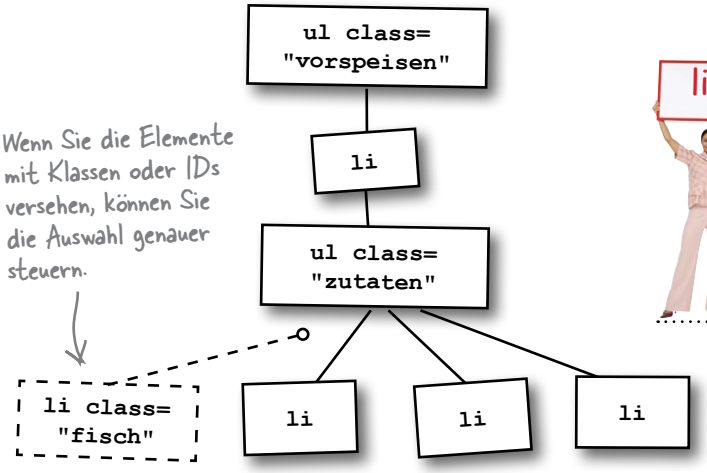
```
$("div#top").remove()
```

Vergessen Sie nicht, dass alle Elemente entfernt werden, auf die der Selektor passt.



```
$("div p").detach()
```

Wenn Sie die Elemente mit Klassen oder IDs versehen, können Sie die Auswahl genauer steuern.



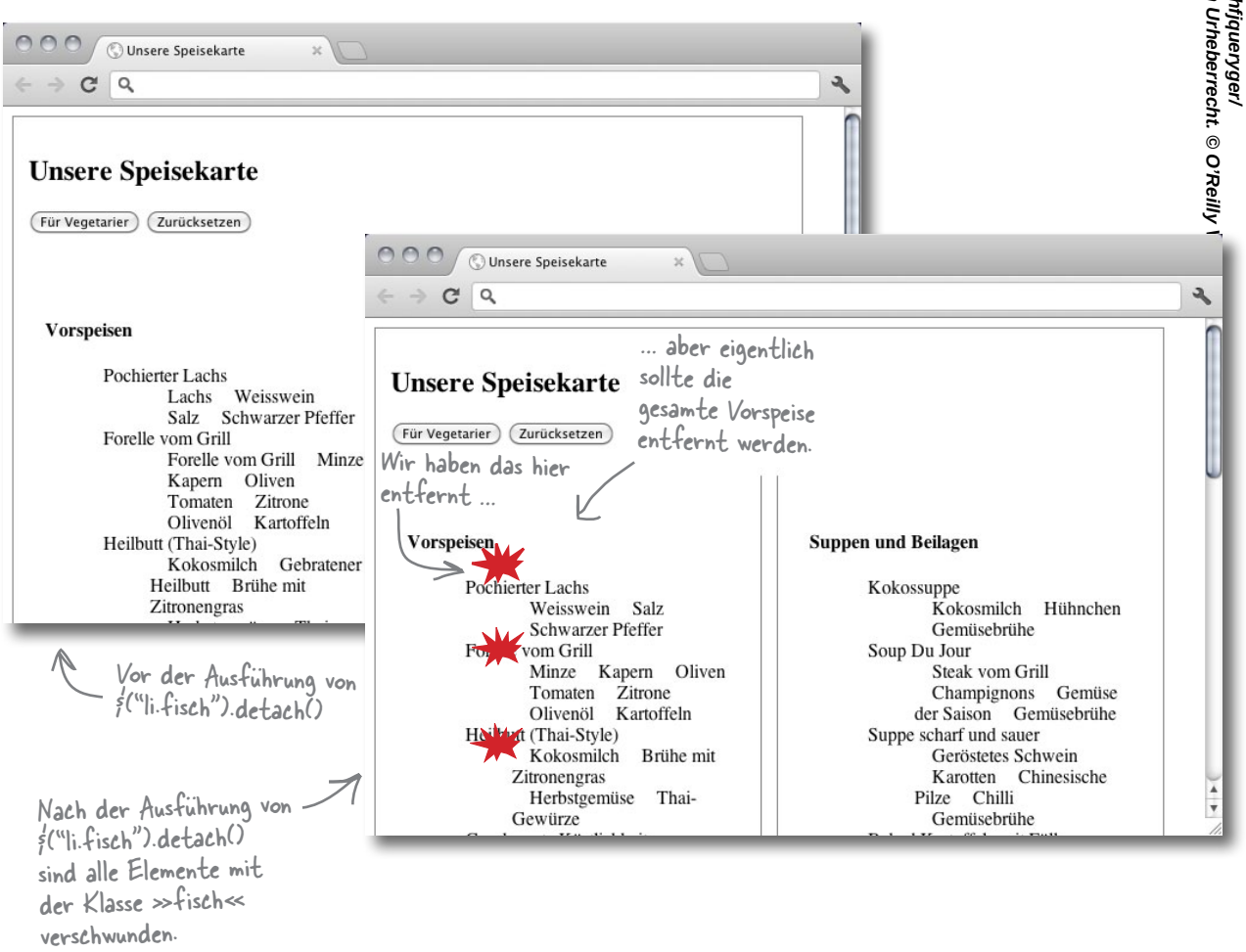
```
$("li.fisch").detach()
```

Dies ist ein Auszug aus dem Buch "jQuery von Kopf bis Fuß", ISBN 978-3-86899-189-5
http://www.oreilly.de/catalog/hfjqueryger/
Dieser Auszug unterliegt dem Urheberrecht. © O'Reilly Verlag 2011

Ryan Benedetti / Ronan Cranley, jQuery von Kopf bis Fuß, O´Reilly, ISBN 9783868991895

PROBEFAHRT

Bauen Sie den Code aus der dritten Lösung in die click-Funktion für den »Für Vegetarier«-Button ein und speichern Sie sie in Ihrer *meine_skripts.js*-Datei. Danach öffnen Sie die Seite im Browser Ihrer Wahl und überprüfen, ob alles wie gewünscht funktioniert.



Die `detach`-Methode hat einiges entfernt, aber noch nicht alles, was wir loswerden wollten. Eigentlich müssen wir die gesamte Vorspeise entfernen, die innerhalb des Listeneintrags mit der Klasse »fisch« verschachtelt ist.

Wie können wir das DOM anweisen, die gesamte Vorspeise zu entfernen?

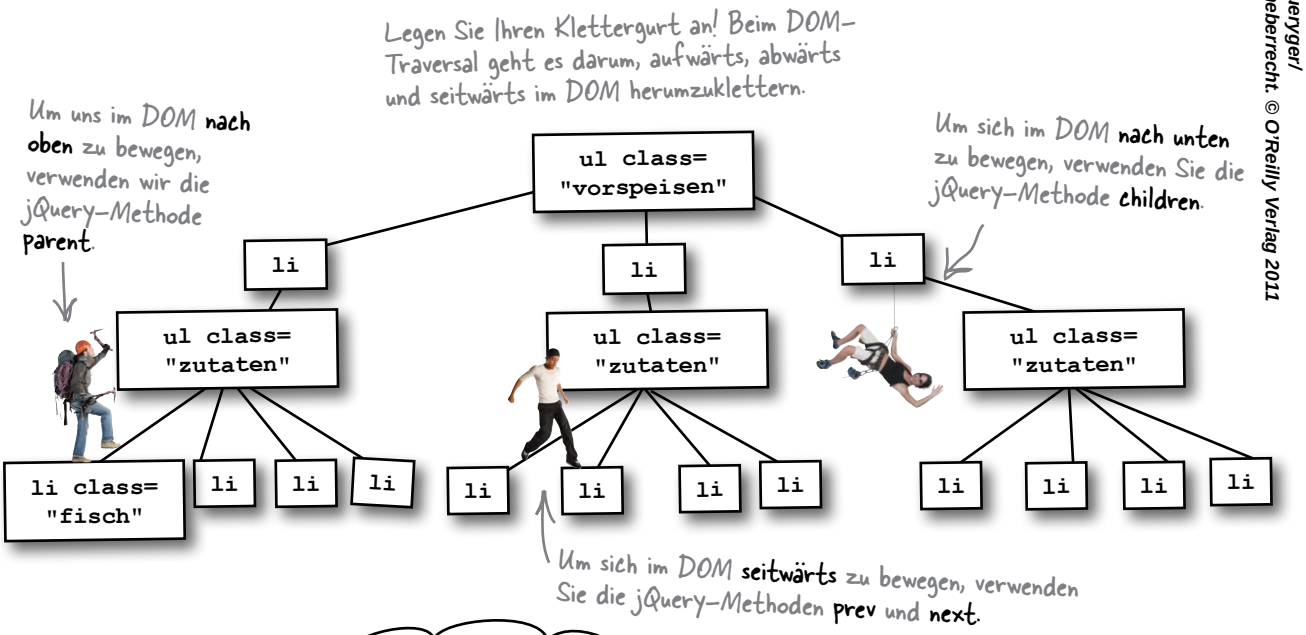
Dies ist ein Auszug aus dem Buch "jQuery von Kopf bis Fuß", ISBN 978-3-86899-189-5
http://www.oreilly.de/catalog/hfjqueryger/
Dieser Auszug unterliegt dem Urheberrecht. © O'Reilly

Im DOM-Baum herumklettern

In Kapitel 1 haben Sie erfahren, dass das DOM wie ein Baum aufgebaut ist: Es besitzt eine Wurzel, Äste und Knoten. Der JavaScript-Interpreter im Browser kann sich im DOM *bewegen* (und es dann verändern) – und besonders gut geht das mit jQuery. Sich im DOM zu bewegen, bedeutet, im DOM-Baum von einem Ast zum anderen auf und ab und hin und her zu klettern.

Im Prinzip haben wir das DOM seit Kapitel 1 immer wieder verändert. Die gerade gezeigte detach-Methode ist ein Beispiel dafür (etwa das Entfernen von Elementen aus dem DOM).

Aber worum geht es bei diesem auf Englisch »Traversal« genannten Konzept? Nehmen wir dafür einen Teil der Speisekarte und visualisieren ihn als DOM-Baum, um zu sehen, wie das Traversal funktioniert.



»Im DOM-Baum herumklettern?« Klar ... Und wie soll diese Kletterei mir bitte beim Entfernen der Vorspeisen helfen?

Durch die Traversal-Methoden können Sie ein Element auswählen, aber tatsächlich mit anderen Elementen daneben, darüber oder darunter arbeiten.

Wie diese Methoden funktionieren, wollen wir uns einmal genauer ansehen.

Ryan Benedetti / Ronan Cranley, jQuery von Kopf bis Fuß, O´Reilly, ISBN 9783868991895

Dies ist ein Auszug aus dem Buch "jQuery von Kopf bis Fuß", ISBN 978-3-86899-189-5
http://www.oreilly.de/catalog/hfjqueryger/
Dieser Auszug unterliegt dem Urheberrecht. © O'Reilly Verlag 2011

Traversal-Methoden bewegen sich im DOM

Um dem DOM mitzuteilen, dass Vorspeisen entfernt werden sollen, die Fisch enthalten, müssen wir die Elemente anhand ihrer Beziehung zueinander angeben. Mit den Traversal-Methoden von jQuery können wir Elemente über ihre Beziehungen zueinander auswählen.

Alle Elemente mit der Klasse >>fisch<< auswählen.

Und dann deren Elternelement verwenden.

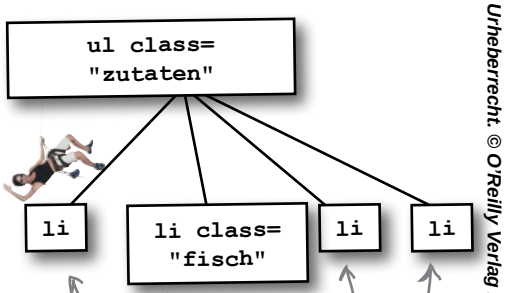
```
$(".fisch").parent()
```

Dies ist das Elternelement von li.fisch.

Alle Elemente mit der Klasse >>zutaten<< auswählen.

Und dann deren Kindelemente verwenden.

```
$(".zutaten").children()
```

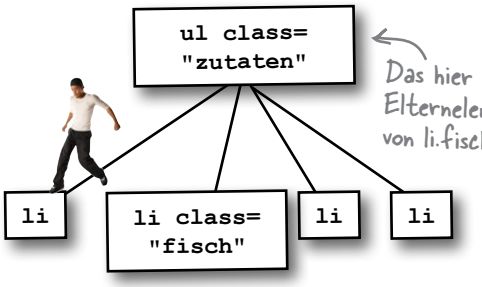


Das hier sind die Kindelemente von ul.zutaten.

Alle Elemente mit der Klasse >>fisch<< auswählen.

Und dann das jeweils links daneben liegende Geschwisterelement verwenden.

```
$(".fisch").prev()
```

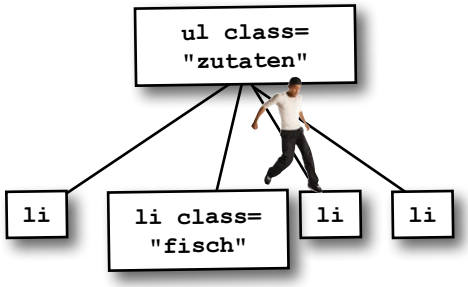


Das hier ist das Elternelement von li.fisch.

Alle Elemente mit der Klasse >>fisch<< auswählen.

Und dann das jeweils rechts daneben liegende Geschwisterelement verwenden.

```
$(".fisch").next()
```



KOPF-NUSS

Welche dieser Methoden kann uns dabei helfen, die Vorspeisen zu entfernen, die Fisch (bzw. »fisch«) enthalten?

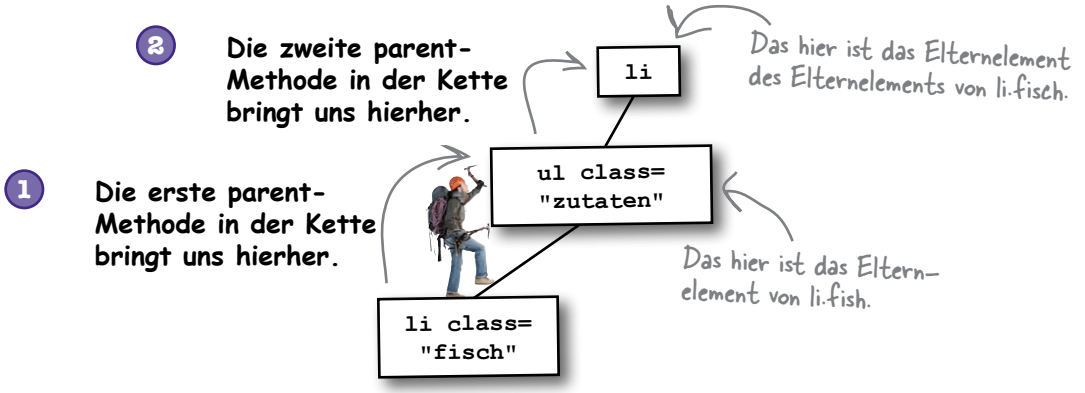
Dies ist ein Auszug aus dem Buch "jQuery von Kopf bis Fuß", ISBN 978-3-86899-189-5
http://www.oreilly.de/catalog/hfjqueryger/
Dieser Auszug unterliegt dem Urheberrecht. © O'Reilly Verlag 2011

Methoden verketten, um weiter zu klettern

Was, wenn wir höher, niedriger oder tiefer klettern wollen? Dann heißt es: Ketten anlegen! jQuery besitzt die Möglichkeit, Methoden zu *verketten*. Durch verkettete Methoden sind effizientere Bewegungen und Änderungen am DOM unserer Seiten möglich. Und das geht so:

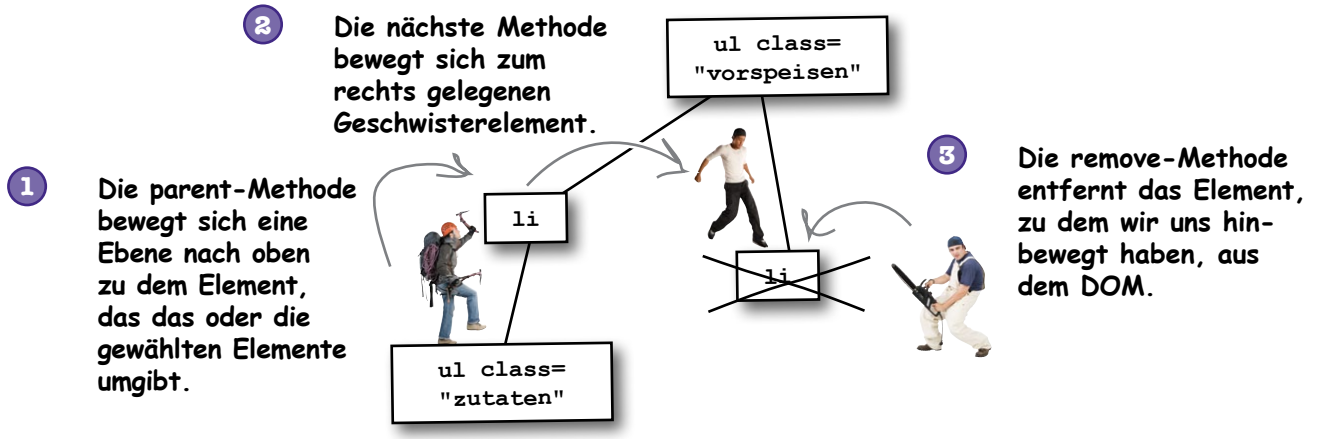
Um zum nächsthöheren Elternelement zu gelangen, erweitern Sie die Methodenkette einfach um ein weiteres Glied.

```
$(".fisch").parent().parent()
```



Sie können die Methoden beliebig miteinander kombinieren.

```
$(".zutaten").parent().next().remove()
```



Ryan Benedetti / Ronan Cranley, jQuery von Kopf bis Fuß, O´Reilly, ISBN 9783868991895

Dies ist ein Auszug aus dem Buch "jQuery von Kopf bis Fuß", ISBN 978-3-86899-189-5
http://www.oreilly.de/catalog/hfjqueryger/
Dieser Auszug unterliegt dem Urheberrecht. © O'Reilly Verlag 2011

Gehen Sie zur Adresse *http://www.thinkjquery.com/chapter04/traversal/* und öffnen Sie die JavaScript-Konsole in den Entwicklerwerkzeugen Ihres Browsers. Im Abschnitt »Lies mich« am Anfang dieses Buches werden verschiedene Entwicklerwerkzeuge vorgestellt. Führen Sie alle vier Traversal-Methoden mit einer verketteten `detach`-Methode aus, wie es unten gezeigt wird. Danach schreiben Sie hier auf, ob die Methode bei der Lösung unseres Problems helfen kann oder nicht. Den eingedeutschten Code finden Sie unter *www.oreilly.de/catalog/hfjqueryger*.

Wichtig: Vergessen Sie nicht, **die Seite *neu zu laden***, nachdem Sie die jeweilige Anweisung ausgeführt haben.

```
$(".vorspeisen").children().detach()
```

.....

.....

.....

```
$(".zutaten").children().detach()
```

.....

.....

.....

```
$(".fisch").parent().detach()
```

.....

.....

.....

```
$(".fisch").parent().parent().detach()
```

.....

.....

.....

Dies ist ein Auszug aus dem Buch "jQuery von Kopf bis Fuß", ISBN 978-3-86899-189-5
http://www.oreilly.de/catalog/hfjqueryger/
Dieser Auszug unterliegt dem Urheberrecht. © O'Reilly Verlag 2011

Lösung zur Übung

Gehen Sie zur Adresse *http://www.thinkjquery.com/chapter04/traversal/* und öffnen Sie die JavaScript-Konsole in den Entwicklerwerkzeugen Ihres Browsers. Im Abschnitt »Lies mich« am Anfang dieses Buches werden verschiedene Entwicklerwerkzeuge vorgestellt. Führen Sie alle vier Traversal-Methoden mit einer verketteten `detach`-Methode aus, wie es unten gezeigt wird. Danach schreiben Sie hier auf, ob die Methode bei der Lösung unseres Problems helfen kann oder nicht. Den eingedeutschten Code finden Sie unter *www.oreilly.de/catalog/hfjqueryger*.

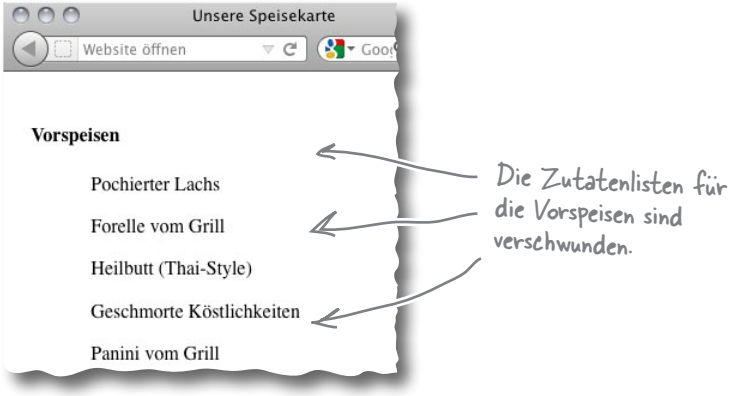
Wichtig: Vergessen Sie nicht, **die Seite *neu zu laden***, nachdem Sie die jeweilige Anweisung ausgeführt haben.

```
$(".vorspeisen").children().detach()
```

Diese Traversal-Methode entfernt die Kindelemente von ul.vorspeisen. Die Methode funktioniert nicht für das Entfernen von >>fisch<<, weil ALLE Vorspeisen verschwinden. Mist! Nicht, was wir brauchen.

```
$(".zutaten").children().detach()
```

Diese Traversal-Methode löscht die Kindelemente von ul.zutaten. Sie funktioniert nicht, um bestimmte Vorspeisen zu entfernen, die >>fisch<< enthalten. Stattdessen werden die Listen der Zutaten abgetrennt. Ups!

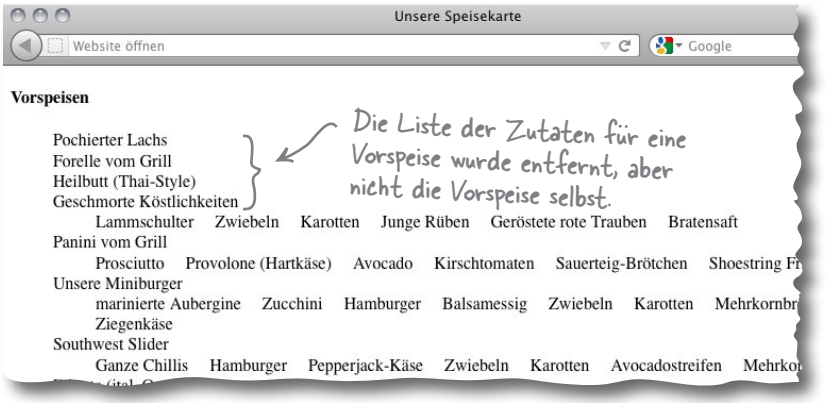


Dies ist ein Auszug aus dem Buch "jQuery von Kopf bis Fuß", ISBN 978-3-86899-189-5
http://www.oreilly.de/catalog/hfjqueryger/
Dieser Auszug unterliegt dem Urheberrecht. © O'Reilly Verlag 2011

Ryan Benedetti / Ronan Cranley, jQuery von Kopf bis Fuß, O´Reilly, ISBN 9783868991895

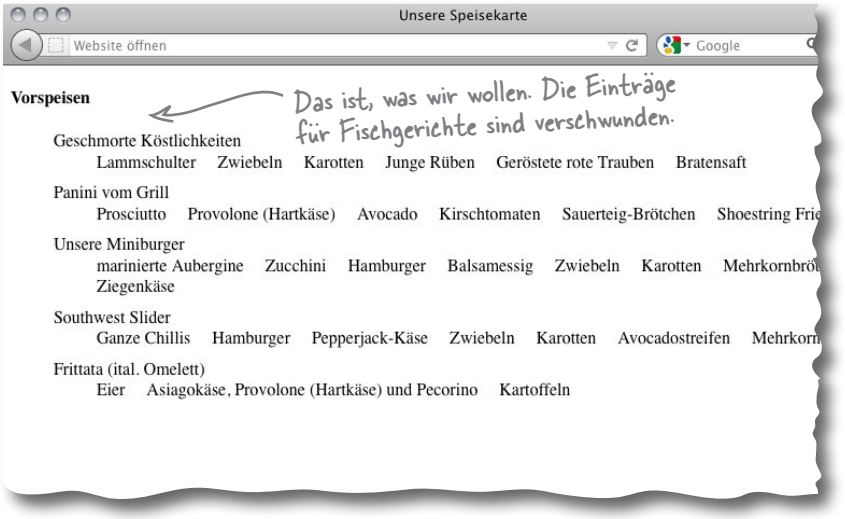
```
$(".fisch").parent().detach()
```

Diese Traversal-Methode entfernt das Elternelement von >>fisch<<. Für das Entfernen von Vorspeisen, die >>fisch<< enthalten, funktioniert sie nicht, weil sie nicht weit genug im DOM-Baum aufsteigt. Stattdessen wird ul.zutaten (und alles darin enthaltene) entfernt.




```
$(".fisch").parent().parent().detach()
```

Diese Traversal-Methode entfernt das Elternelement des Elternelements (sozusagen das >>Großeltern-Element<<) von >>fisch<<. Das ist genau, was wir brauchen.



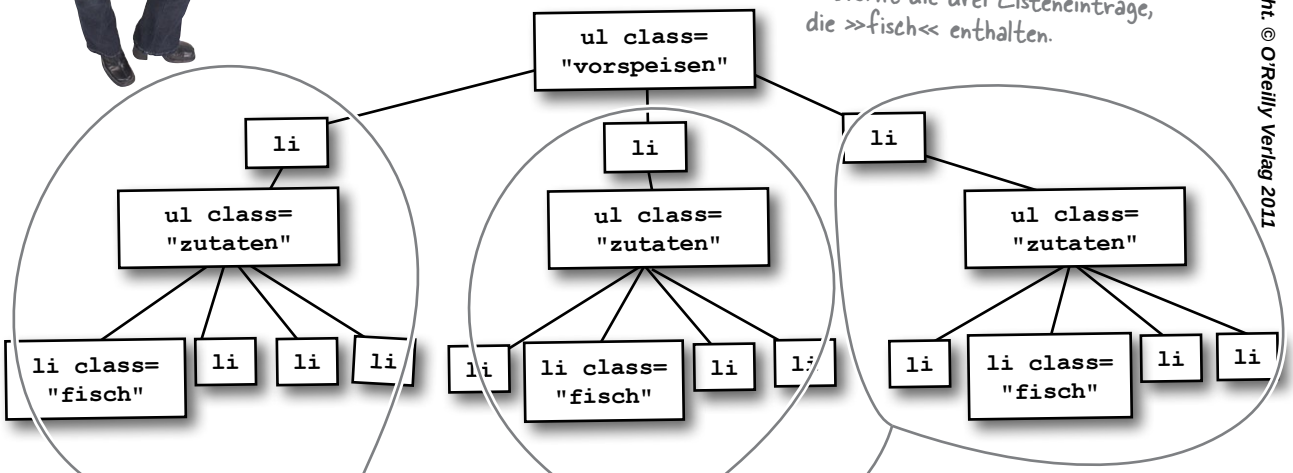
Dies ist ein Auszug aus dem Buch "jQuery von Kopf bis Fuß", ISBN 978-3-86899-189-5
http://www.oreilly.de/catalog/hfjqueryger/
Dieser Auszug unterliegt dem Urheberrecht. © O'Reilly Verlag 2011

Moment mal! Müssen wir die Fisch-Vorspeisen nicht wieder anzeigen, wenn jemand den »Zurücksetzen«-Button anklickt?

Richtig. Wir können die Fisch-Einträge nicht einfach enfernen und dann vergessen.

Wir müssen unseren Code noch einmal überdenken, damit das funktioniert.

$("fisch").parent().parent().detach() entfernt die drei Listeneinträge, die »fisch« enthalten.



Wir können die Elemente mit detach-Methode aus dem DOM entfernen, aber wie bekommen wir sie bei Bedarf wieder hinein?

KOPF-NUSS

Wir müssen die Fischgerichte später wieder anzeigen. Wie könnte das gehen?

Ryan Benedetti / Ronan Cranley, jQuery von Kopf bis Fuß, O´Reilly, ISBN 9783868991895

Dies ist ein Auszug aus dem Buch "jQuery von Kopf bis Fuß", ISBN 978-3-86899-189-5
http://www.oreilly.de/catalog/hfjqueryger/
Dieser Auszug unterliegt dem Urheberrecht. © O'Reilly Verlag 2011

Spitzen Sie Ihren Bleistift

Mittlerweile haben wir schon eine ganze Menge jQuery- und JavaScript-Konstrukte gesehen. Welche davon könnten wir benutzen, damit die Elemente mit der Klasse ».fisch« nicht vergessen werden? Schreiben Sie entweder »Ja« oder »Nein« in die Spalte »Benutzen oder nicht?« und erklären Sie, warum Sie das Konstrukt gewählt bzw. nicht gewählt haben. Eine Antwort haben wir schon für Sie gegeben. Sie brauchen also nur noch drei Mal zu antworten.

	Benutzen oder nicht?	Warum?
Terminator	Nein	Ein Terminator beendet eine Anweisung. Dadurch können wir uns nicht an die entfernten Elemente erinnern.
Variable		
Funktion		
Selektor		

Dies ist ein Auszug aus dem Buch "jQuery von Kopf bis Fuß", ISBN 978-3-86899-189-5
http://www.oreilly.de/catalog/hfjqueryger/
Dieser Auszug unterliegt dem Urheberrecht. © O'Reilly Verlag 2011

Spitzen Sie Ihren Bleistift
Lösung

Mittlerweile haben wir schon eine ganze Menge jQuery- und JavaScript-Kontrukte gesehen. Welche davon könnten wir benutzen, damit die Elemente mit der Klasse ».fisch« nicht vergessen werden? Schreiben Sie entweder »Ja« oder »Nein« in die Spalte »Benutzen oder nicht?« und erklären Sie, warum Sie das Konstrukt gewählt bzw. nicht gewählt haben. Hier ist unsere Lösung.

	Benutzen oder nicht?	Warum?
Terminator	Nein	Ein Terminator beendet eine Anweisung. Dadurch können wir uns nicht an die entfernten Elemente erinnern.
Variable	Ja	Eine Variable kann Dinge für uns speichern. Wenn wir die entfernten Elemente speichern, können wir sie später einfach zurückholen, indem wir den Inhalt der Variablen auslesen.
Funktion	Nein	Mit einer Funktion können wir Daten verändern. Das Problem mit den entfernten Elementen hat aber mit der Speicherung von Daten zu tun, nicht mit ihrer Veränderung.
Selektor	Nein	Ein Selektor wählt ein Element anhand der Dinge im DOM aus. Wir haben unsere Elemente bereits ausgewählt. Wir müssen diese Elemente jetzt speichern.

Es gibt keine Dummen Fragen

F: Ich kenne remove und detach. Was mache ich, wenn ich nicht das Element, sondern nur seinen Inhalt entfernen will?

A: Um den Inhalt eines Elements zu entfernen, können Sie die `empty`-Methode verwenden. Um den Inhalt eines Absatzes auf einer Seite zu entfernen, können Sie beispielsweise diese Formulierung verwenden: `$("p").empty();`.

F: Gibt es eine Möglichkeit, sämtliche Vorfahrenelemente eines Elements auszuwählen?

A: Ja. Zusätzlich zur `parent`-Methode besitzt jQuery die Methode `parents`. Damit können Sie sämtliche Vorfahren des gewählten Elements ansprechen. Weiter hinten in diesem Kapitel können Sie die Methode in Aktion erleben.

F: Was kann ich machen, wenn ich das nächstgelegene Vorfahrenelement eines bestimmten Typs finden möchte?

A: Dafür können Sie die `closest`-Methode benutzen. Wie die `parents`-Methode steigt `closest` im DOM-Baum nach oben. Sobald sie einen Treffer erzielt, hält sie an. Wenn Sie beispielsweise für ein li-Element das nächstgelegene ul-Element finden möchten, können Sie schreiben: `$("li").closest("ul")`.

F: Ich kenne die Methoden `next` und `previous`. Was kann ich tun, um alle Elemente auf der gleichen Ebene im DOM-Baum zu finden?

A: Glücklicherweise haben die jQuery-Entwickler auch daran gedacht. Die Methode `siblings` findet alle Geschwisterelemente auf der gleichen Ebene wie das ausgewählte Element.

F: Ist jQuery in Google Chrome bereits eingebaut?

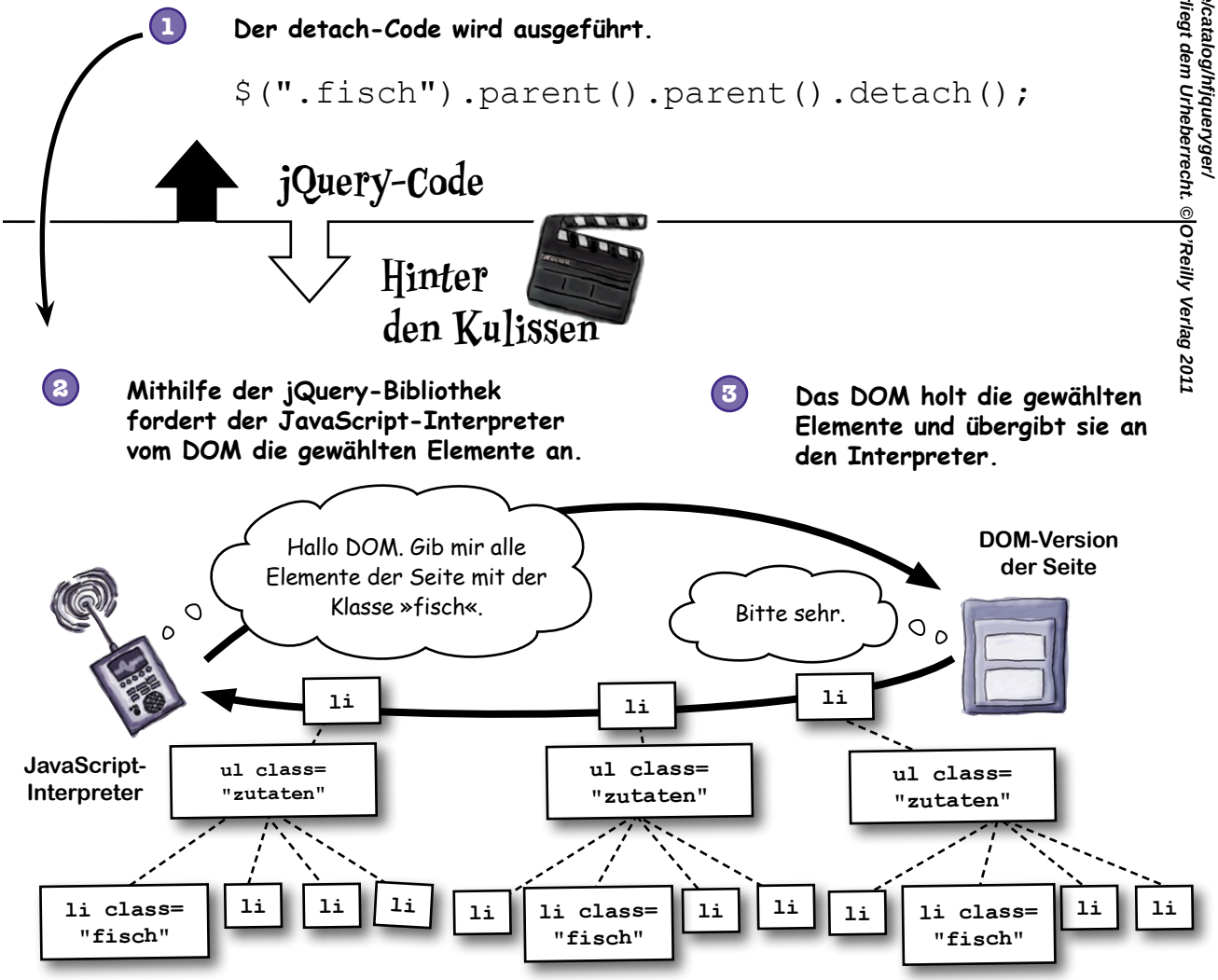
A: Der Grund dafür, dass Sie jQuery in den Entwicklerwerkzeugen von Chrome benutzen können, liegt darin, dass wir jQuery am Anfang der HTML-Seite eingebettet haben. Wenn Sie eine Webseite besuchen, auf der jQuery nicht benutzt wird, sollten Sie nicht erwarten, dass die JavaScript-Konsole von Chrome jQuery-Anweisungen ausführt.

Ryan Benedetti / Ronan Cranley, jQuery von Kopf bis Fuß, O´Reilly, ISBN 9783868991895

Dies ist ein Auszug aus dem Buch "jQuery von Kopf bis Fuß", ISBN 978-3-86899-189-5
http://www.oreilly.de/catalog/hfjqueryger/
Dieser Auszug unterliegt dem Urheberrecht. © O'Reilly Verlag 2011

Speichern von Elementen in Variablen

Variablen sind offensichtlich ziemlich praktisch. Immerhin benutzen wir sie immer wieder. Wir haben Variablen bereits in den ersten drei Kapiteln gesehen. Bisher haben wir sie aber nur verwendet, um Zahlen und Text zu speichern. Wäre es nicht bequem, wenn Variablen auch unsere Elemente speichern könnten?



Der Browser bewahrt die Elemente kurze Zeit im Arbeitsspeicher auf. Wenn wir sie später im Programm noch einmal benutzen möchten, ist es eine gute Idee, sie in Variablen zu speichern. Aber wie geht das?

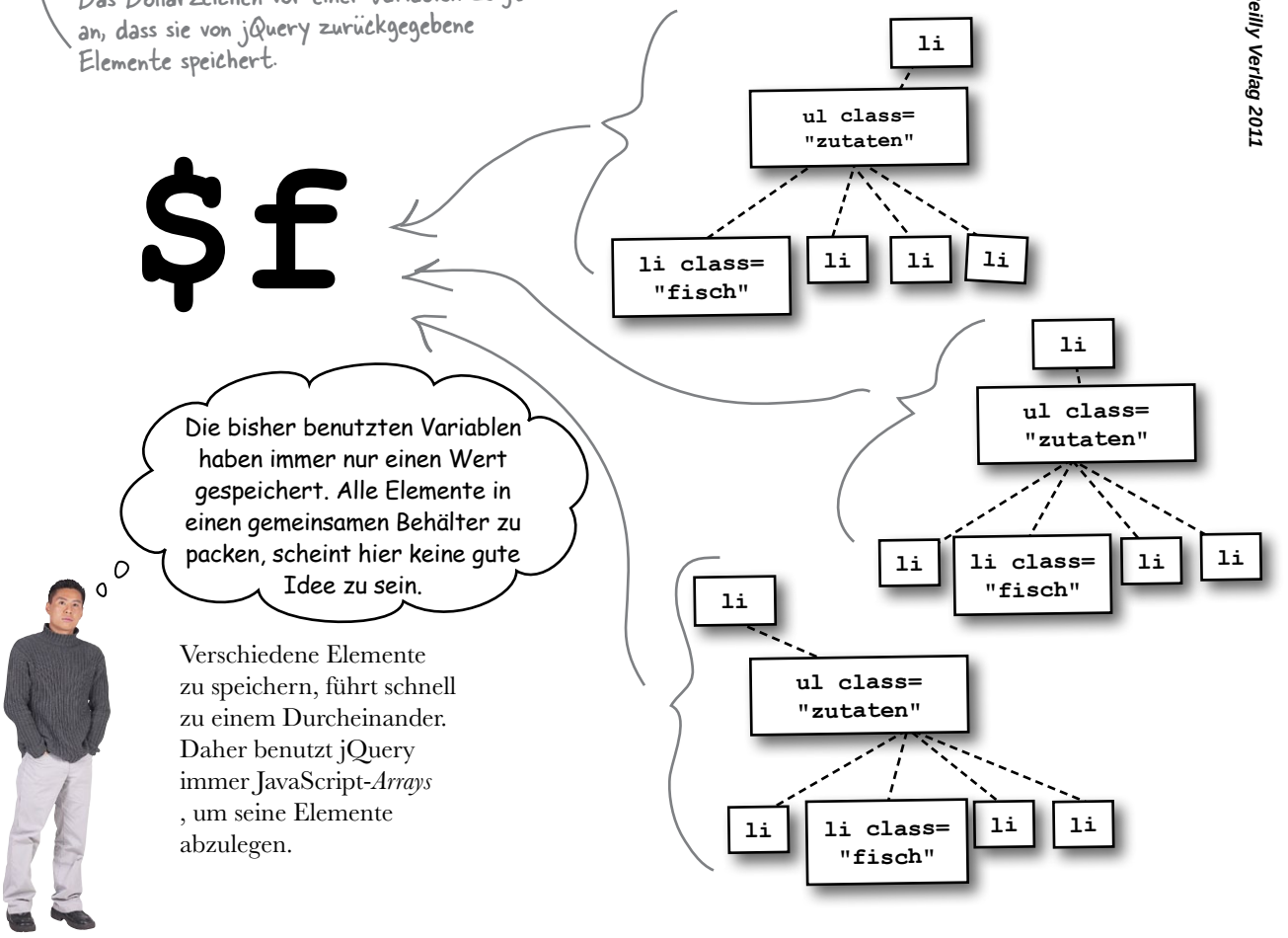
Dies ist ein Auszug aus dem Buch "jQuery von Kopf bis Fuß", ISBN 978-3-86899-189-5
http://www.oreilly.de/catalog/hfjqueryger/
Dieser Auszug unterliegt dem Urheberrecht. ©O'Reilly Verlag 2011

Schon wieder dieses Dollarzeichen ...

Das Speichern der Elemente ist einfach. Zunächst erzeugen wir eine Variable, genau wie beim Speichern von Text und Zahlen. Dann weisen wir ihr (mithilfe des Gleichheitszeichens) das Ergebnis der Anweisung zu, die die Elemente zurückgibt. Aber eigentlich wäre es gut, zu wissen, wenn die Variable *besondere Dinge* (im Gegensatz zu einfachem Text oder Zahlen) speichert, oder? Üblicherweise stellen jQuery-Coder Variablen, die zurückgegebene Elemente speichern, ein Dollarzeichen voran. Auf diese Weise lässt sich schnell erkennen, dass in der Variablen Dinge von jQuery gespeichert werden.

```
$f = $(".fisch").parent().parent().detach();
```

Das Dollarzeichen vor einer Variablen zeigt an, dass sie von jQuery zurückgegebene Elemente speichert.



Verschiedene Elemente zu speichern, führt schnell zu einem Durcheinander. Daher benutzt jQuery immer JavaScript-*Arrays*, um seine Elemente abzulegen.

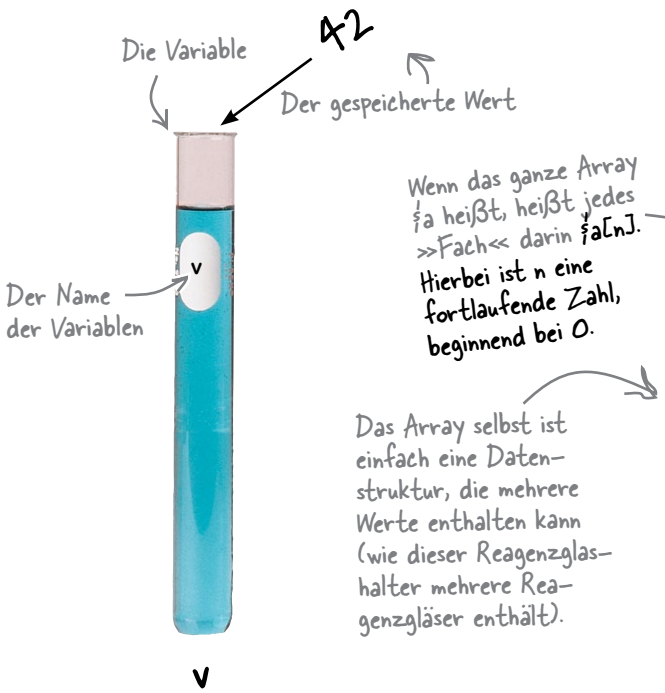
Dies ist ein Auszug aus dem Buch "jQuery von Kopf bis Fuß", ISBN 978-3-86899-189-5
http://www.oreilly.de/catalog/hfjqueryger/
Dieser Auszug unterliegt dem Urheberrecht. © O'Reilly Verlag 2011

Ryan Benedetti / Ronan Cranley, jQuery von Kopf bis Fuß, O´Reilly, ISBN 9783868991895

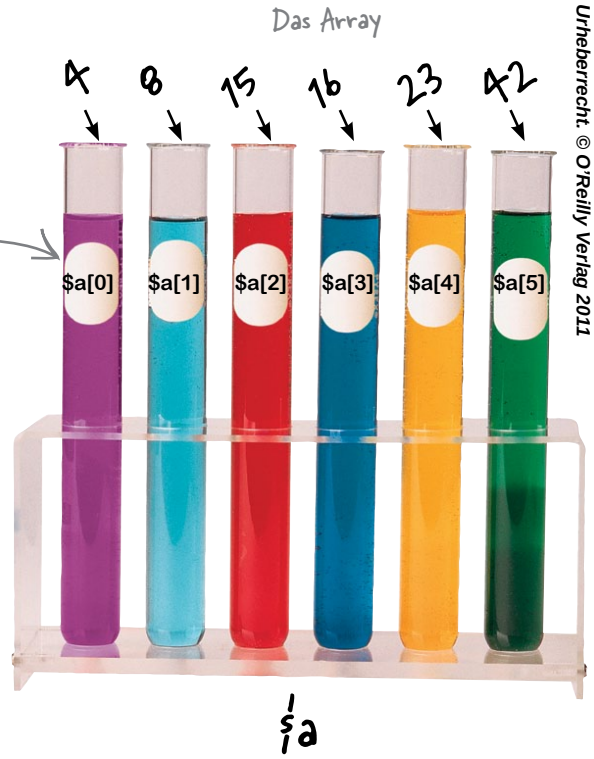
Die Speichermöglichkeiten mit Arrays erweitern

Immer wenn wir Elemente aus dem DOM auswählen und in einer Variable speichern, gibt jQuery diese Daten als *Array* zurück. Ein Array ist einfach eine Variable mit größeren Speichermöglichkeiten.

Eine einfache Variable speichert einen Wert.



Ein Array speichert mehrere Werte.



Wenn das ganze Array $a heißt, heißt jedes »Fach« darin $a[n]. Hierbei ist n eine fortlaufende Zahl, beginnend bei 0.

Das Array selbst ist einfach eine Datenstruktur, die mehrere Werte enthalten kann (wie dieser Reagenzglashalter mehrere Reagenzgläser enthält).

Wir können jedes Fach mit Daten befüllen und sie wieder auslesen. Um den Wert »15« im dritten Fach abzulegen, würden wir das hier schreiben:

```
$a[2] = 15;
```

Das dritte Fach hat die Nummer 2, weil wir bei 0 angefangen haben.

Freak-Futter

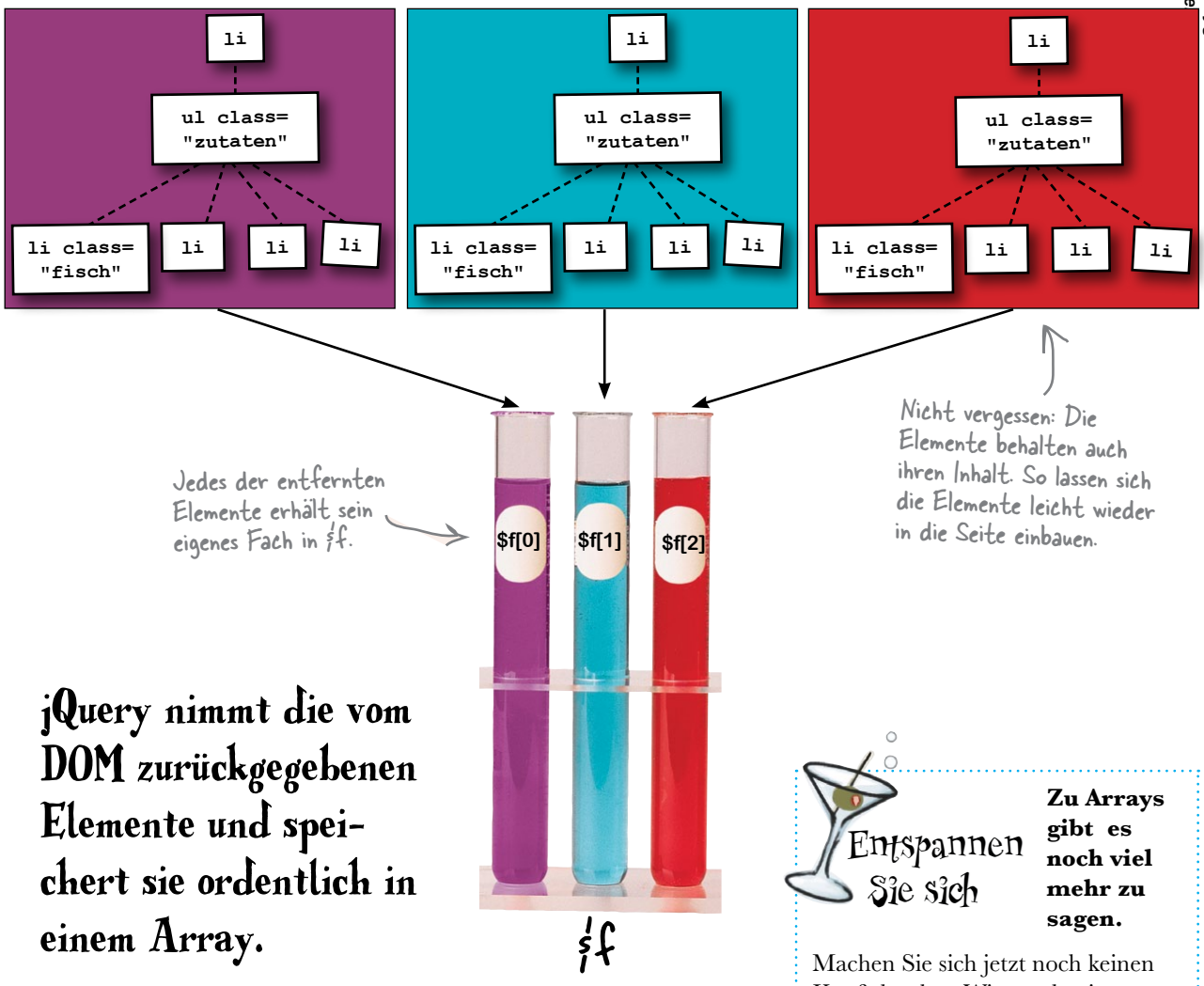
Arrays **müssen** nicht mit einem Dollarzeichen ($) beginnen. Die Praxis, jQuery-Arrays ein Dollarzeichen voranzustellen ist eine Programmierkonvention von jQuery-Entwicklern.

Dies ist ein Auszug aus dem Buch "jQuery von Kopf bis Fuß", ISBN 978-3-86899-189-5
http://www.oreilly.de/catalog/hfjqueryger/
Dieser Auszug unterliegt dem Urheberrecht. © O'Reilly Verlag 2011

Elemente in einem Array speichern

Wenn wir die li-Elemente auswählen, entfernen und einer Variablen (`$f`) zuweisen, nimmt jQuery die vom DOM zurückgegebenen Elemente und speichert sie ordentlich in einem JavaScript-Array. Dadurch wird die Arbeit nicht so schwierig, wenn jemand den »Zurücksetzen«-Button anklickt und die Elemente wieder an ihren alten Platz müssen.

```
$f = $(".fisch").parent().parent().detach();
```



jQuery nimmt die vom DOM zurückgegebenen Elemente und speichert sie ordentlich in einem Array.

Entspannen Sie sich

Zu Arrays gibt es noch viel mehr zu sagen.

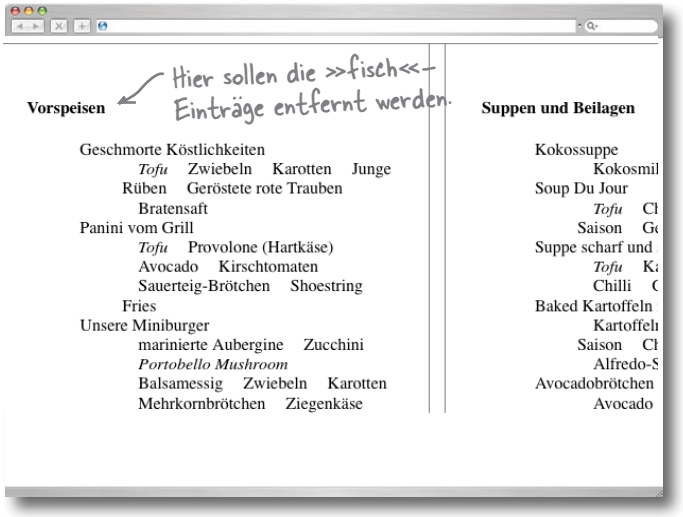
Machen Sie sich jetzt noch keinen Kopf darüber. Wir werden in Kapitel 6 viel genauer darauf eingehen.

Dies ist ein Auszug aus dem Buch "jQuery von Kopf bis Fuß", ISBN 978-3-86899-189-5
http://www.oreilly.de/catalog/hfjqueryger/
Dieser Auszug unterliegt dem Urhe

Ryan Benedetti / Ronan Cranley, jQuery von Kopf bis Fuß, O´Reilly, ISBN 9783868991895

PROBEFAHRT

Auf der vorigen Seite steht eine Codezeile, die die »Großeltern« der »fisch«-Elemente entfernt und dem Array $f zuweist. Bauen Sie diese Zeile in die click-Funktion für den `vegetarisch_an`-Button in der *meine_skripts.js*-Datei ein. Danach öffnen Sie die Seite in Ihrem Browser, um zu testen, ob alles nach Wunsch funktioniert.



Sie haben's geschafft! Zeit, die Aufgabenliste zu aktualisieren.

- [x] 1. Finde Elemente mit der Klasse >>fisch<< und entferne sie aus der Liste.
- [ ] 2. Finde li-Elemente mit der Klasse >>hamburger<< und ersetze sie durch Portobello-Champignons.
- [ ] 3. Finde li-Elemente mit der Klasse >>fleisch<< und ersetze sie durch Tofu.

Als Nächstes müssen Sie die Vorspeisen finden, die Hamburger enthalten, und den Hamburger durch Portobello-Champignons ersetzen.

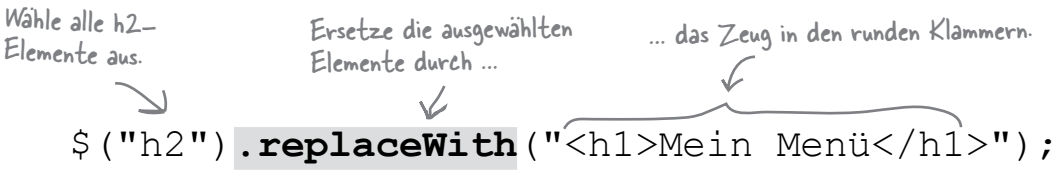
KOPF-NUSS

Mittlerweile wissen wir, wie man Elemente aus dem DOM entfernt. Aber wie können wir den Inhalt von DOM-Elementen dynamisch durch andere Inhalte ersetzen?

Dies ist ein Auszug aus dem Buch "jQuery von Kopf bis Fuß", ISBN 978-3-86899-189-5
http://www.oreilly.de/catalog/hfjqueryger/
Dieser Auszug unterliegt dem Urheberrecht. © O'Reilly Verlag 2011

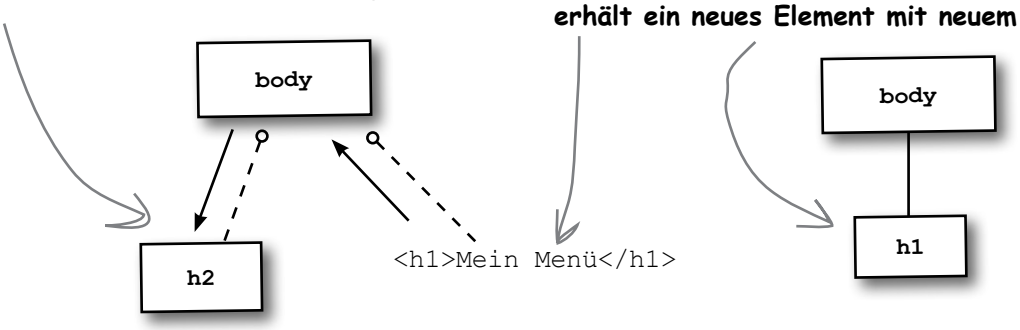
Elemente mit replaceWith verändern

Mit der Methode `replaceWith` ist es möglich, ausgewählte Elemente durch andere zu ersetzen. Diese praktische jQuery-Methode können Sie immer dann benutzen, wenn Sie etwas im DOM gegen etwas anderes austauschen wollen. Angenommen, Sie möchten die h2-Überschrift mit dem Inhalt »Unser Menü« durch eine Überschrift erster Ordnung mit dem Inhalt »Mein Menü« ersetzen. Dann können Sie dafür die `replaceWith`-Methode verwenden, wie hier gezeigt wird:



1 **Der JS-Interpreter findet eine Überschrift zweiter Ordnung ...**

2 **... und ersetzt sie durch das, was innerhalb der runden Klammern steht. Das DOM erhält ein neues Element mit neuem Inhalt.**



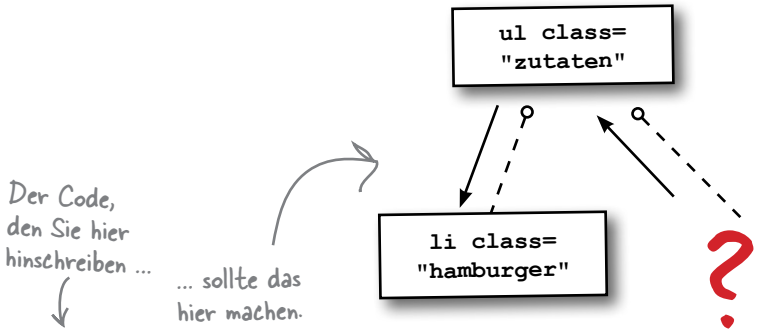
Dies ist ein Auszug aus dem Buch "jQuery von Kopf bis Fuß", ISBN 978-3-86899-189-5
http://www.oreilly.de/catalog/hfjqueryger/
Dieser Auszug unterliegt dem Urheberrecht. © O'Reilly Verlag 2011

Ryan Benedetti / Ronan Cranley, jQuery von Kopf bis Fuß, O´Reilly, ISBN 9783868991895

Wie kann replaceWith uns helfen?

Es sollen li-Elemente mit der Klasse »hamburger« gefunden und durch ein `li`-Element mit der Klasse »portobello« ersetzt werden. Lassen Sie uns kurz über das Problem nachdenken, bevor wir unseren Code schreiben.

Schreiben Sie den Code, der `li`-Elemente mit der Klasse »hamburger« findet und durch `li`-Elemente mit der Klasse »portobello« ersetzt. Das unten stehende Diagramm soll Ihnen beim Finden der Lösung helfen. Einen Teil der Antwort haben wir schon für Sie aufgeschrieben. Den Rest machen Sie selbst.

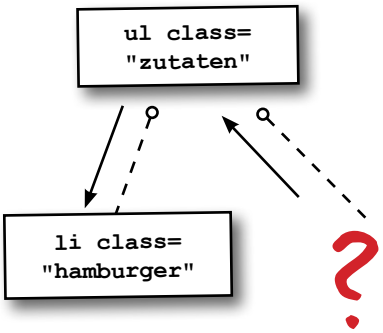



```
$(..................).replaceWith(.......................................<em>Portobello-Champignons</em>.........);
```

Dies ist ein Auszug aus dem Buch "jQuery von Kopf bis Fuß", ISBN 978-3-86899-189-5
http://www.oreilly.de/catalog/hfjqueryger/
Dieser Auszug unterliegt dem Urheberrecht. © O'Reilly Verlag 2011

Lösung zur Übung

Schreiben Sie den Code, der `li`-Elemente mit der Klasse »hamburger« findet und durch `li`-Elemente mit der Klasse »portobello« ersetzt. Hier ist unsere Lösung.



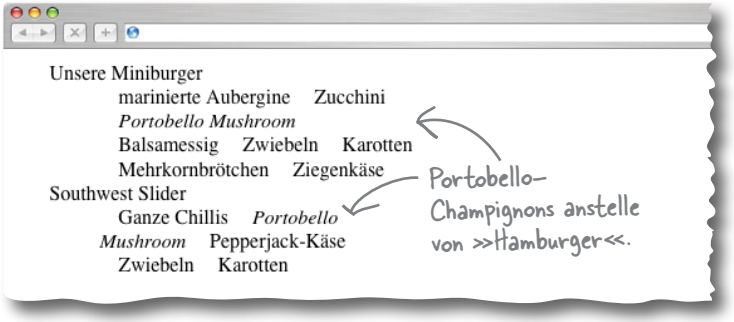
Wählt alle Elemente mit der Klasse "hamburger" aus.

Die replaceWith-Methode tauscht den gewählten Inhalt dynamisch durch das Element in den runden Klammern aus. Vergessen Sie nicht, dass die Klammern auch HTML-Code enthalten dürfen.

```
$(".hamburger").replaceWith("<li class='portobello'><em>Portobello-Champignons</em> </li>");
```

PROBEFAHRT

Fügen Sie den `replaceWith`-Code in die Button-click-Funktion `vegetarisch_an` in Ihrer *meine_skripts.js*-Datei ein. Öffnen Sie die Seite dann in Ihrem Browser und betätigen Sie den »Vegetarisch«-Button, um sich zu vergewissern, dass alles läuft wie gewünscht.



Ryan Benedetti / Ronan Cranley, jQuery von Kopf bis Fuß, O´Reilly, ISBN 9783868991895

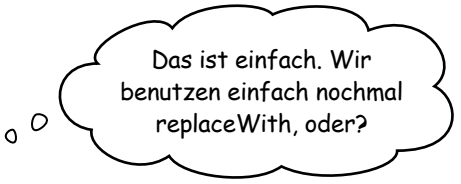
Dies ist ein Auszug aus dem Buch "jQuery von Kopf bis Fuß", ISBN 978-3-86899-189-5
http://www.oreilly.de/catalog/hfjqueryger/
Dieser Auszug unterliegt dem Urheberrecht. © O'Reilly Verlag 2011

Denken Sie voraus, bevor Sie replaceWith verwenden

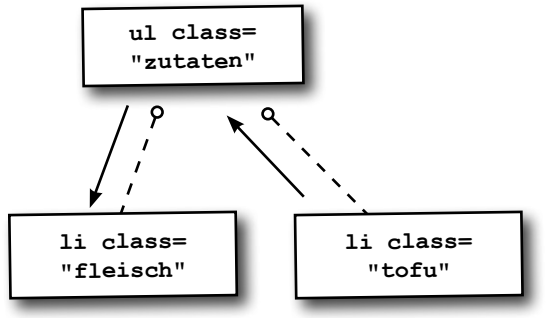
Was steht als nächstes auf der Checkliste?

- [x] 1. Finde Elemente mit der Klasse »fisch« und entferne sie aus der Liste.
- [x] 2. Finde li-Elemente mit der Klasse »hamburger« und ersetze sie durch Portobello-Champignons.
- [ ] 3. Finde li-Elemente mit der Klasse »fleisch« und ersetze sie durch Tofu.

Sie müssen die Vorspeisen mit der Klasse »fleisch« durch Tofu ersetzen.



Nein, wir können `replaceWith` hier nicht einsetzen.



KOPF-NUSS

Die jQuery-Methode `replaceWith` ist einfach und kann eine Menge. Leider kann sie uns bei diesem Problem nicht weiterhelfen. Warum nicht?

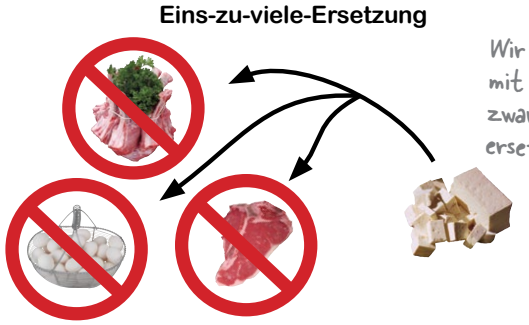
Dies ist ein Auszug aus dem Buch "jQuery von Kopf bis Fuß", ISBN 978-3-86899-189-5
http://www.oreilly.de/catalog/hfjqueryger/
Dieser Auszug unterliegt dem Urheberrecht. © O'Reilly Verlag 2011

replaceWith funktioniert nicht in jeder Situation

Die `replaceWith`-Methode funktioniert gut, wenn es um Eins-zu-eins-Ersetzungen geht, beispielsweise wenn die »hamburger«-Klasse durch die »portobello«-Klasse ausgetauscht werden soll.

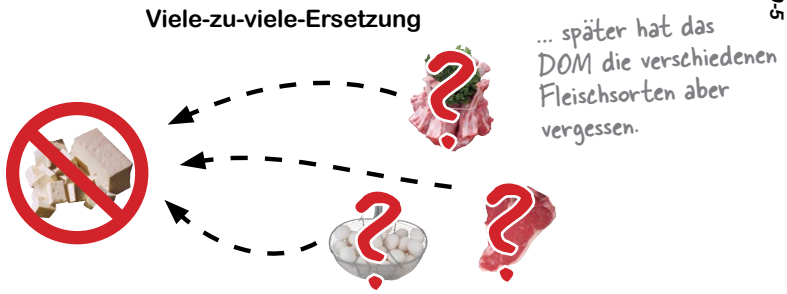
Eins-zu-viele-Ersetzung

Die Eins-zu-eins-Methode funktioniert beim nächsten Punkt auf unserer Checkliste aber nicht. Hier müssen wir *mehrere* verschiedene Zutaten (z. B. Truthahn, Eier, Steak, Lammfleisch) durch eine Zutat (Tofu) ersetzen.



Viele-zu-viele-Ersetzung

Aber wenn wir später die »tofu«-Elemente auswählen und ersetzen wollen, haben wir ein Problem. Sobald wir die verschiedenen Fleischsorten wieder einbauen wollen, hat das DOM sie alle vergessen. Wir könnten »tofu« durch *eine* Fleischsorte ersetzen, aber das wollen wir ja nicht. Daher müssen wir diese Ersetzung in der Speisekarte in zwei Schritten durchführen:



1. **Fügen Sie die li-Elemente mit der Klasse »tofu« nach den »fleisch«-Elementen ein.**
2. **Entfernen Sie die Elemente mit der Klasse »fleisch« und speichern Sie sie in einer Variablen.**

Ryan Benedetti / Ronan Cranley, jQuery von Kopf bis Fuß, O´Reilly, ISBN 9783868991895

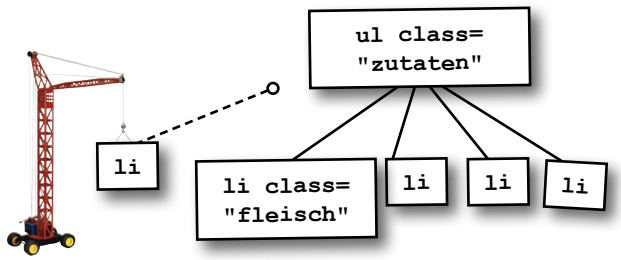
Dies ist ein Auszug aus dem Buch "jQuery von Kopf bis Fuß", ISBN 978-3-86899-189-5
http://www.oreilly.de/catalog/hfjqueryger/
Dieser Auszug unterliegt dem Urheberrecht. © O'Reilly Verlag 2011

HTML-Inhalte in das DOM einfügen

Bisher haben wir DOM-Elemente nur entfernt oder ersetzt. Zum Glück haben sich die Entwickler der jQuery-Bibliothek auch Möglichkeiten ausgedacht, um Sachen in das DOM einzufügen. Wir meinen die Methoden `before` und `after`.

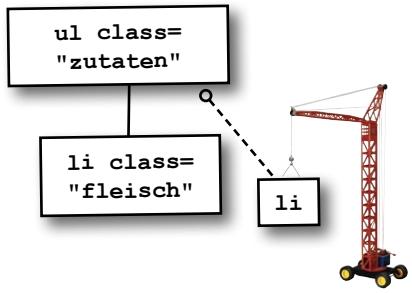
`before` fügt Inhalte vor dem gewählten Element ein und `after` danach.

```
$(".fleisch").before("<li>Tofu</li>");
```



`after` fügt Inhalte nach dem ausgewählten Element ein.

```
$(".fleisch").after("<li>Tofu</li>");
```



Spitzen Sie Ihren Bleistift

Schreiben Sie den jQuery-Code, der die Schritte unserer Lösung ausführt.

1. **Fügen Sie die li-Elemente mit der Klasse »tofu« nach den »fleisch«-Elementen ein.** .....

2. **Entfernen Sie die Elemente mit der Klasse »fleisch« und speichern Sie sie in einer Variablen.** .....

Dies ist ein Auszug aus dem Buch "jQuery von Kopf bis Fuß", ISBN 978-3-86899-189-5
http://www.oreilly.de/catalog/hfjqueryger/
Dieser Auszug unterliegt dem Urheberrecht. © O'Reilly Verlag 2011

Spitzen Sie Ihren Bleistift
Lösung

Schreiben Sie den jQuery-Code, der die Schritte unserer Lösung ausführt.

1. **Fügen Sie die li-Elemente mit der Klasse »tofu« nach den »fleisch«-Elementen ein.**

```
$(".fleisch").after("<li class='tofu'><em>Tofu</em></li>");
```

2. **Entfernen Sie die Elemente mit der Klasse »fleisch« und speichern Sie sie in einer Variablen.**

```
$m = $(".fleisch").detach();
```

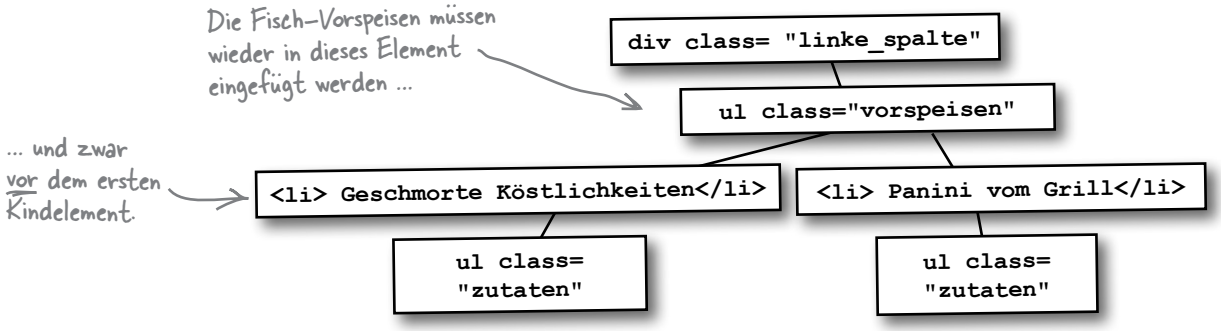
Damit haben Sie alle Anforderungen auf unserer Checkliste erfüllt:

- [x] 1. Finde Elemente mit der Klasse »fisch« und entferne sie aus der Liste.
- [x] 2. Finde li-Elemente mit der Klasse »hamburger« und ersetze sie durch Portobello-Champignons.
- [x] 3. Finde li-Elemente mit der Klasse »fleisch« und ersetze sie durch Tofu.

Als Nächstes müssen wir den »Menü zurücksetzen«-Button erstellen. Hier kommt das, was der Button tun soll:

- [ ] Fügen Sie die Fisch-Vorspeisen wieder an ihrer ursprünglichen Position im DOM ein (also vor dem ersten Listeneintrag in der linken Spalte).
- [ ] Finden Sie Vorspeisen mit Portobello-Champignons und ersetzen Sie sie durch Hamburger.
- [ ] Finden Sie die Vorspeisen, die Tofu enthalten, und ersetzen Sie sie (in der richtigen Reihenfolge) durch die verschiedenen Fleischsorten.

Lassen Sie uns gleich loslegen und sehen, was wir für die erste Anforderung brauchen.



Wir wissen, wir man `before` benutzt, aber wie geben wir das *erste* Kindelement an?

Ryan Benedetti / Ronan Cranley, jQuery von Kopf bis Fuß, O´Reilly, ISBN 9783868991895

Dies ist ein Auszug aus dem Buch "jQuery von Kopf bis Fuß", ISBN 978-3-86899-189-5
http://www.oreilly.de/catalog/hfjqueryger/
Dieser Auszug unterliegt dem Urheberrecht. © O'Reilly Verlag 2011

Die Auswahl mit Filtermethoden einschränken (Teil 1)

Glücklicherweise bietet jQuery eine Reihe von Filtermethoden, mit denen sich Probleme wie das Finden des ersten Kindelements lösen lassen. Sechs davon wollen wir uns hier ansehen (drei auf dieser Seite, drei auf der nächsten).

first

Die `first`-Methode filtert aus den ausgewählten Elementen *alles außer dem ersten Element* heraus.

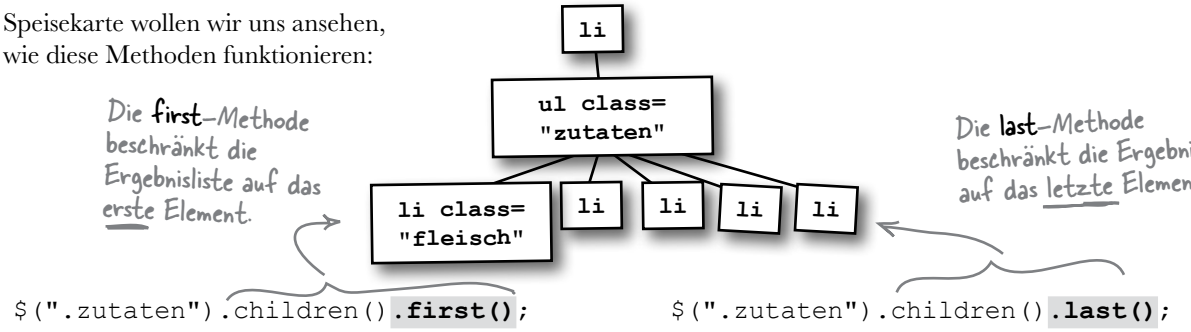
eq

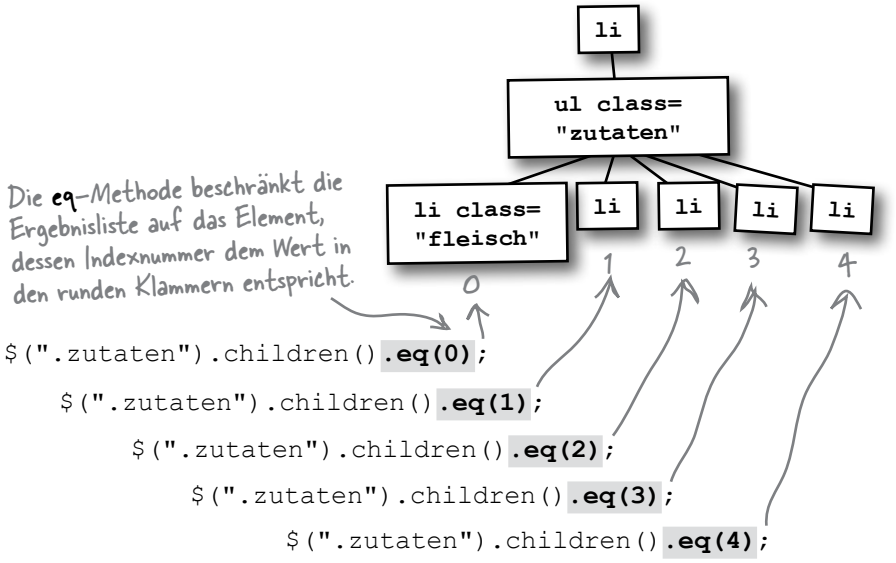
Die `eq`-Methode filtert aus den gewählten Elementen *alles heraus, was nicht die index-Nummer* in den runden Klammern hat.

last

Die `last`-Methode filtert aus den gewählten Elementen *alles außer dem letzten Element* heraus.

Anhand eines Gerichts auf der Speisekarte wollen wir uns ansehen, wie diese Methoden funktionieren:





Vergessen Sie nicht, dass jQuery die ausgewählten Elemente als Array zurückgibt. Die Indexnummer in den runden Klammern der eq-Methode entspricht dem Fach des Arrays.

Dies ist ein Auszug aus dem Buch "jQuery von Kopf bis Fuß", ISBN 978-3-86899-189-5
http://www.oreilly.de/catalog/hfjqueryger/
Dieser Auszug unterliegt dem Urheberrecht. © O'Reilly Verlag 2011

Die Auswahl mit Filtermethoden einschränken (Teil 2)

Und jetzt wollen wir einen Blick auf die Methoden `slice`, `filter` und `not` und ihre Funktionsweisen werfen.

slice

Die `slice`-Methode filtert aus der Ergebnisliste alle Elemente aus, deren Indexnummern nicht *zwischen den Indexnummern liegen, die in den runden Klammern stehen.*

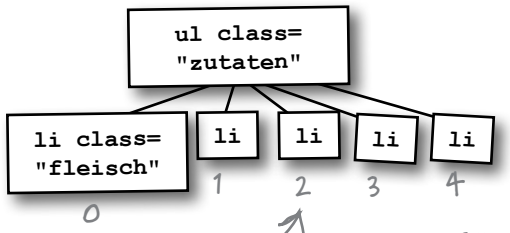
filter

Die `filter`-Methode filtert alle Elemente aus der Ergebnisliste, bis auf diejenigen, *auf die der Selektor* in den runden Klammern passt.

not

Die `not`-Method filtert alle Elemente aus der Ergebnisliste, auf die *der Selektor in den runden Klammern* nicht passt.

Mit der **slice**–Methode werden werden nur die ausgewählten Elemente benutzt, deren Indexnummern zwischen den Werten liegen, die in den runden Klammern angegeben wurden.

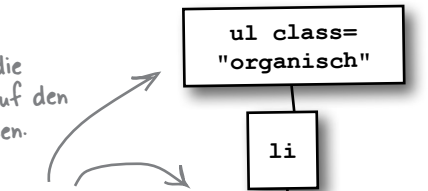



```
$(".zutaten").children().slice(1,3);
```

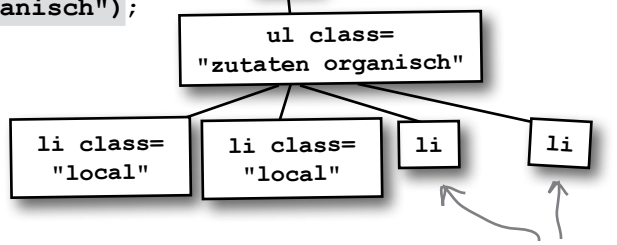
In diesem Fall wird nur das zweite li–Element zurückgegeben.

Mit den Methoden `filter` und `not` können Sie anhand der als Argumente übergebenen Selektoren eine Untermenge der ausgewählten Elemente erstellen.

Bei der **filter**–Methode werden nur die ausgewählten Elemente benutzt, die auf den Selektor in den runden Klammern passen.




```
$(".zutaten").parents().filter(".organisch");
```



Die Methoden filter und not lassen sich gut mit den Methoden **parents** und **children** verwenden.

Mit der parents –Methode können wir Elemente finden, die Eltern, Großeltern, Urgroßeltern usw. des gewählten Elements sind.

```
$("ul.zutaten.organisch").children().not(".local");
```

Die **not**–Methode beschränkt die ausgewählten Elemente auf diejenigen, die nicht auf den in den runden Klammern angegebenen Selektor passen.

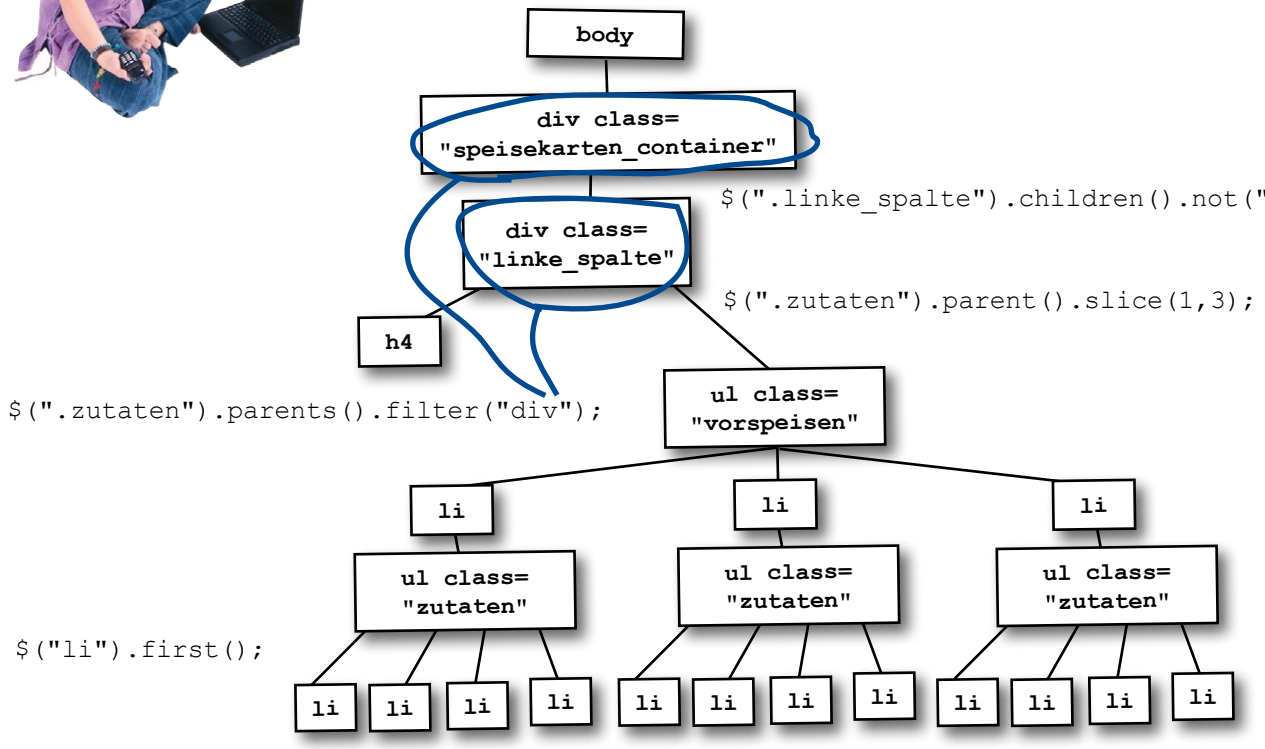
Welche dieser Methoden hilft Ihnen damit, das erste Kindelement in der Speisekarte zu finden?

Ryan Benedetti / Ronan Cranley, jQuery von Kopf bis Fuß, O´Reilly, ISBN 9783868991895

Dies ist ein Auszug aus dem Buch "jQuery von Kopf bis Fuß", ISBN 978-3-86899-189-5
http://www.oreilly.de/catalog/hfjqueryger/
Dieser Auszug unterliegt dem Urheberrecht. © O'Reilly Verlag 2011

SEIEN Sie das DOM

Ihre Aufgabe ist es, selbst DOM zu spielen. Zeichnen Sie eine Linie von der jQuery-Anweisung zu den Elementen im DOM, die der Selektor zurückgibt. Gehen Sie davon aus, dass das die einzigen Elemente der Seite sind. Die erste Verbindung haben wir schon für Sie gezeichnet.




```
$(".linke_spalte").children().not("h4");
$(".zutaten").parent().slice(1,3);
$(".zutaten").parents().filter("div");
$("li").first();
$(".zutaten li").eq(3);
$(".zutaten").children().last();
```

Spitzen Sie Ihren Bleistift

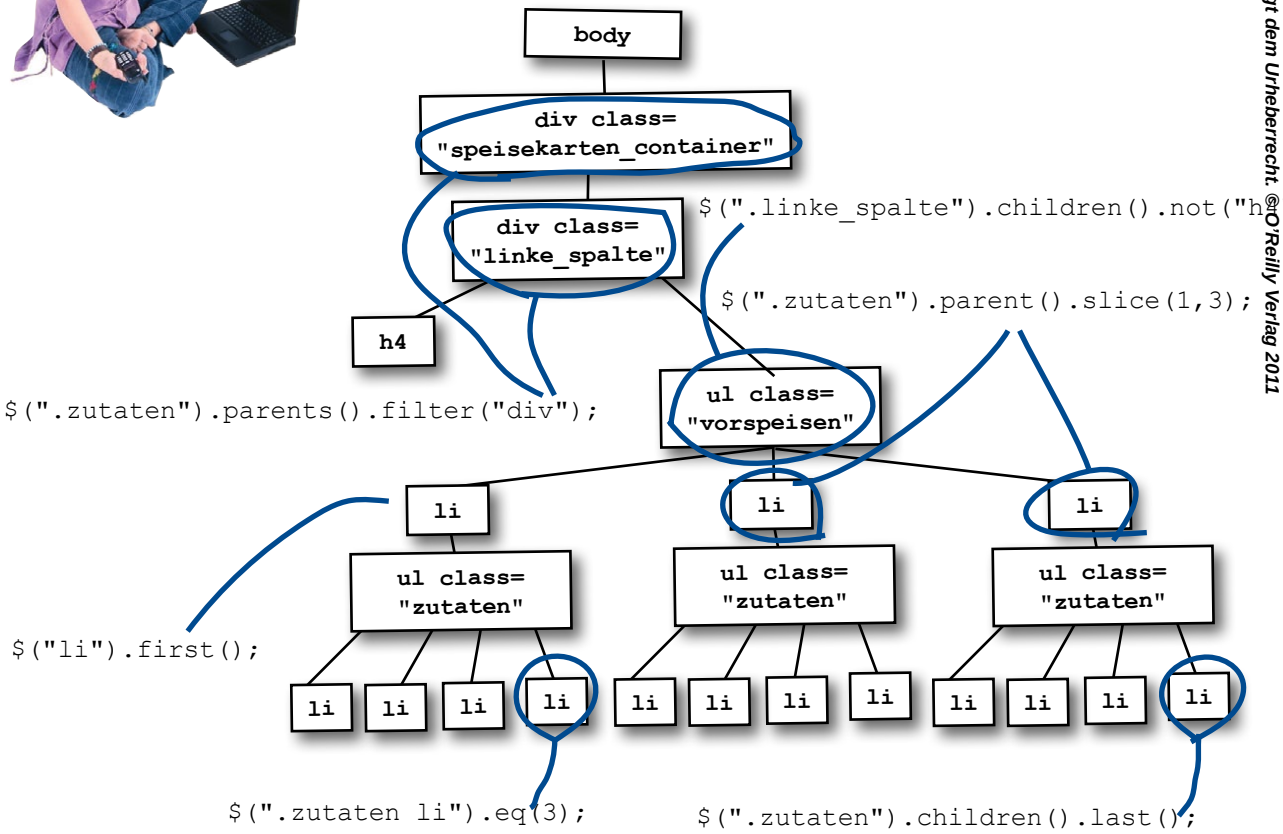
Schreiben Sie die Zeile jQuery-Code auf, die alle Fisch-Vorspeisen wieder an ihre ursprüngliche Stelle in der Speisekarte einfügt (z. B. das erste Gericht unterhalb von »vorspeisen«).

.....before($f);

Dies ist ein Auszug aus dem Buch "jQuery von Kopf bis Fuß", ISBN 978-3-86899-189-5
http://www.oreilly.de/catalog/hfjqueryger/
Dieser Auszug unterliegt dem Urheberrecht. © O'Reilly Verlag 2011

Seien Sie das DOM: Lösung

Ihre Aufgabe ist es, selbst DOM zu spielen. Zeichnen Sie eine Linie von der jQuery-Anweisung zu den Elementen im DOM, die der Selektor zurückgibt. Gehen Sie davon aus, dass das die einzigen Elemente der Seite sind. Die erste Verbindung haben wir schon für Sie gezeichnet.



Dies ist ein Auszug aus dem Buch "jQuery von Kopf bis Fuß", ISBN 978-3-86899-189-5
http://www.oreilly.de/catalog/hfjqueryger/
Dieser Auszug unterliegt dem Urheberrecht. ©O'Reilly Verlag 2011

Spitzen Sie Ihren Bleistift
Lösung

Schreiben Sie die Zeile jQuery-Code auf, die alle Fisch-Vorspeisen wieder an ihre ursprüngliche Stelle in der Speisekarte einfügt (z. B. das erste Gericht unterhalb von »vorspeisen«).

```
$(".vorspeisen li").first()    .before($f);
```

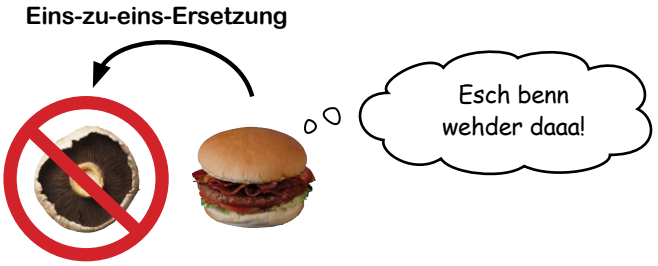

Ryan Benedetti / Ronan Cranley, jQuery von Kopf bis Fuß, O´Reilly, ISBN 9783868991895

Der Hamburger kommt zurück!

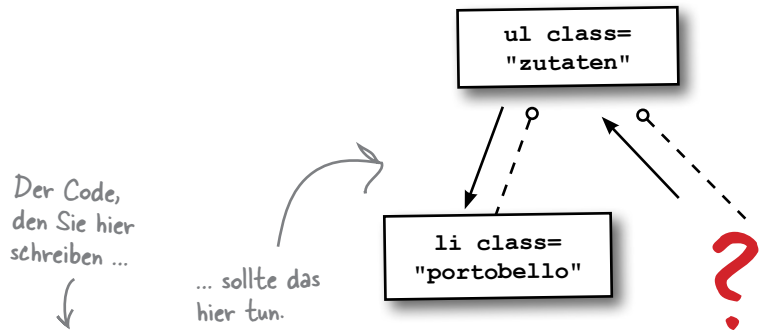
Damit ist die erste Anforderung für den »Zurücksetzen«-Button erfüllt. Zwei haben wir noch:

- [x] Fügen Sie die Fisch-Vorspeisen wieder an ihrer ursprünglichen Position im DOM ein (also vor dem ersten Listeneintrag in der linken Spalte).
- [ ] Finden Sie Vorspeisen mit Portobello-Champignons und ersetzen Sie sie durch Hamburger.
- [ ] Finden Sie die Vorspeisen, die Tofu enthalten, und ersetzen Sie sie (in der richtigen Reihenfolge) durch die verschiedenen Fleischsorten.

Unser nächster Punkt auf der Checkliste kommt einem ein bisschen vor, wie ein Déjà-vu, oder? Im Prinzip müssen wir die ursprüngliche Ersetzung nur umkehren. Warum? Weil wir es hier mit einer Eins-zu-eins-Ersetzung zu tun haben. Und die haben wir besonders gern, weil sie so einfach sind.



Erinnern Sie sich an diese Übung? Wir werden sie einfach umdrehen. Schreiben Sie den Code auf, der alle `li`-Elemente mit der »portobello«-Klasse findet und durch `li`-Elemente der »hamburger«-Klasse ersetzt. Das unten stehende Diagramm soll Ihnen beim Finden der Lösung helfen. Einen Teil der Antwort haben wir schon für Sie aufgeschrieben. Der Rest ist für Sie.

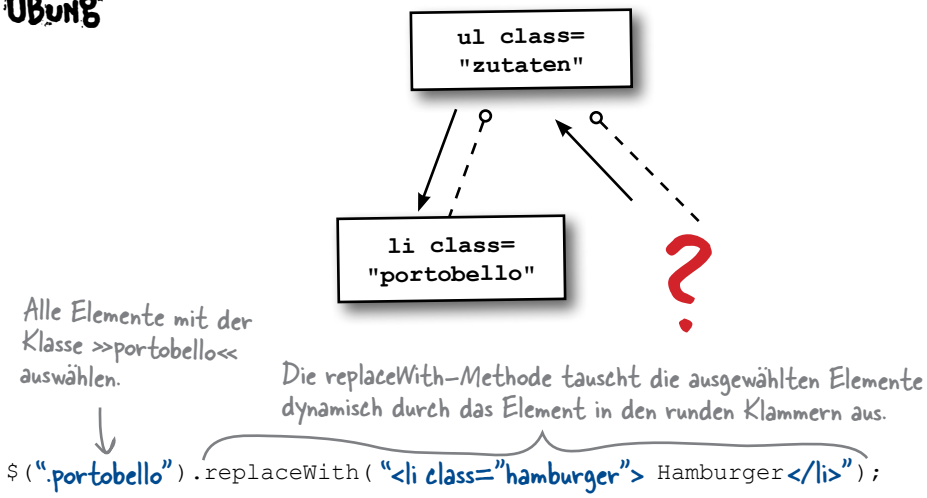



```
$(.....).replaceWith(.....Hamburger.....);
```


Dies ist ein Auszug aus dem Buch "jQuery von Kopf bis Fuß", ISBN 978-3-86899-189-5
http://www.oreilly.de/catalog/hfjqueryger/
Dieser Auszug unterliegt dem Urheberrecht. © O'Reilly Verlag 2011

LÖSUNG ZUR ÜBUNG

`replaceWith` bringt alles in einem Rutsch wieder an Ort und Stelle. Gut gemacht!



Wo ist das Rind (äh, Fleisch)?

Für den »Zurücksetzen«-Button fehlt uns jetzt nur noch ein Punkt auf der Checkliste:

- [x] Fügen Sie die Fisch-Vorspeisen wieder an ihrer ursprünglichen Position im DOM ein (also vor dem ersten Listeneintrag in der linken Spalte).
- [x] Finden Sie Vorspeisen mit Portobello-Champignons und ersetzen Sie sie durch Hamburger.
- [ ] Finden Sie die Vorspeisen, die Tofu enthalten, und ersetzen Sie sie (in der richtigen Reihenfolge) durch die verschiedenen Fleischsorten.

Was haben wir noch mal mit den li-Elementen gemacht? Ein kurzer Blick zurück:

Wir haben die `li.tofu`-Elemente nach den Elementen mit der Klasse »Fleisch« eingefügt.

```
$(".fleisch").after("<li class='tofu'><em>Tofu</em></li>");
```

Wir haben die li.fleisch-Elemente aus dem DOM entfernt und in `$m` gespeichert.

```
$m = $(".fleisch").detach();
```

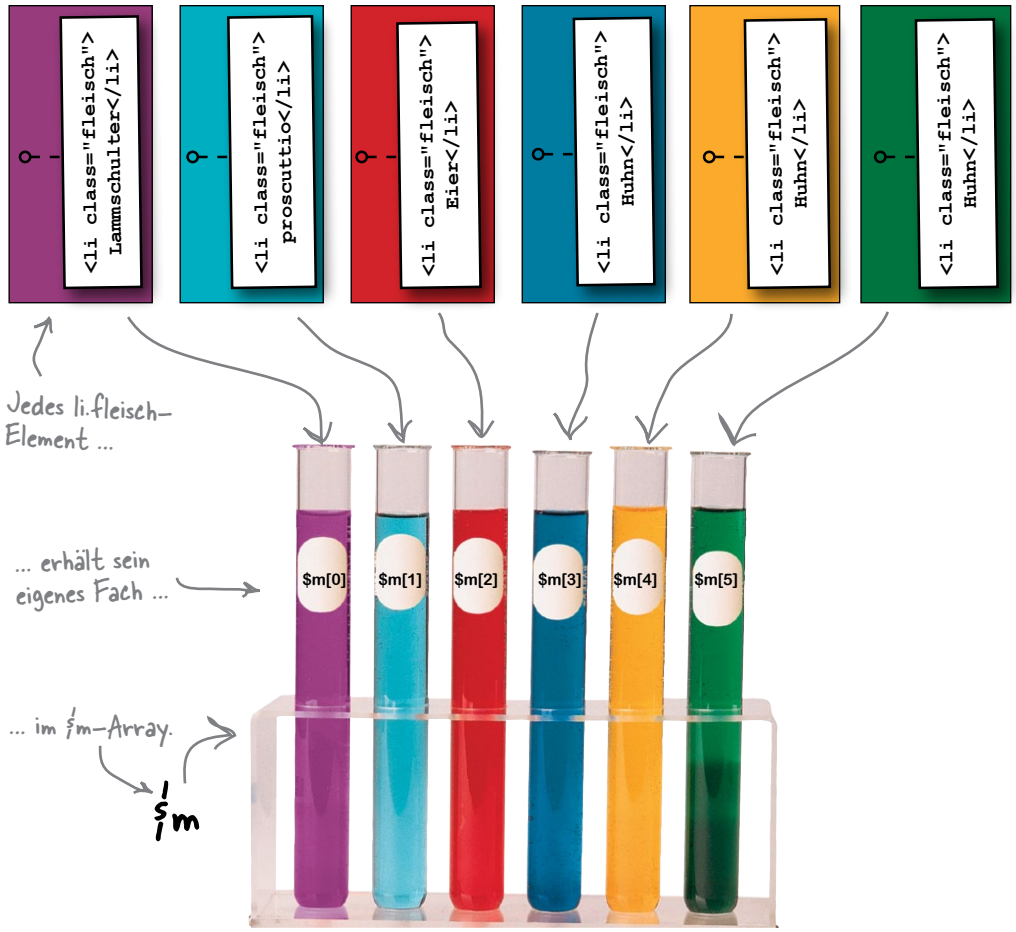
Und wo sind diese Elemente jetzt? Und wie bekommen wir sie zurück?

Ryan Benedetti / Ronan Cranley, jQuery von Kopf bis Fuß, O´Reilly, ISBN 9783868991895

Dies ist ein Auszug aus dem Buch "jQuery von Kopf bis Fuß", ISBN 978-3-86899-189-5
http://www.oreilly.de/catalog/hfjqueryger/
Dieser Auszug unterliegt dem Urheberrecht. © O'Reilly Verlag 2011

Ein fleischiges Array

Wie Sie wissen, erhält die Variable beim Speichern von jQuery-Elementen ein Dollarzeichen. Damit wird angezeigt, dass die verwendete Variable eine besondere Art der Speicherung verwendet. In unserem Fall ist das ein jQuery-Array. Hier sehen Sie, wie die Elemente in `$m` gepeichert wurden:



KOPF-NUSS

Sie müssen die einzelnen `li.fleisch`-Elemente wieder an die Stelle der `li.tofu`-Elemente zurückschieben. Sie haben bereits viele Methoden kennengelernt, um Elemente wieder ins DOM einzubauen. Welche Methode würden Sie in dieser Situation verwenden?

Dies ist ein Auszug aus dem Buch "jQuery von Kopf bis Fuß", ISBN 978-3-86899-189-5
http://www.oreilly.de/catalog/hfjqueryger/
Dieser Auszug unterliegt dem Urheberrecht. © O'Reilly Verlag 2011

Die each-Methode kann eine Schleife über ein Array ausführen.

In Kapitel 3 haben Sie gesehen, wie man die `each`-Methode verwenden kann, um eine Schleife über mehrere Elemente auszuführen. Das können wir auch hier verwenden, um eine Schleife über sämtliche »fleisch«-Elemente im Array `$m` zu erzeugen und sie wieder an ihrem ursprünglichen Ort zu platzieren. Dafür müssen wir uns die Funktionsweise von `each` aber noch etwas genauer ansehen.

Die each-Methode funktioniert wie ein Fließbandroboter für Ihre Elemente.

Der Index (oder Iterator) sorgt dafür, dass wir wissen, an welchem Element die Funktion gerade arbeitet.

i = 0

Richtig interessant wird es, wenn wir innerhalb von each eine Funktion platzieren, die mit jedem Element etwas anstellt, während es verarbeitet wird.

Mit der each-Methode lassen sich jQuery-Aufgaben automatisieren, indem ein Element im Array nach dem anderen abgearbeitet wird.

Das Schlüsselwort this bezieht sich auf das Element, mit dem wir gerade arbeiten.

Wenn wir p-Element auswählen, speichert jQuery die Auswahl in einem Array.

Die each-Methode verarbeitet ein Array-Element nach dem anderen und führt für jedes Element eine bestimmte Aktion aus.

Die Variable i beginnt bei 0 und zählt alle Elemente, die verarbeitet werden.

```
$(".tofu").each(function(i){
    $(this).after(     );
});
```

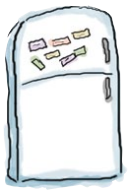
Mit $(this) können Sie der Funktion mitteilen, welches Element gerade verarbeitet wird.

Hier benutzen wir die after-Methode. Im Prinzip können Sie aber jede jQuery-Methode verwenden, um bestimmte Aktionen für ein Array mit Elementen durchzuführen.

Wir wollen die »`fleisch`«-Elemente anstelle der einzelnen `li.tofu`-Elemente einfügen. Was muss hier also stehen?

Dies ist ein Auszug aus dem Buch "jQuery von Kopf bis Fuß", ISBN 978-3-86899-189-5
http://www.oreilly.de/catalog/hfjqueryger/
Dieser Auszug unterliegt dem Urheberrecht. © O'Reilly Verlag 2011

Ryan Benedetti / Ronan Cranley, jQuery von Kopf bis Fuß, O´Reilly, ISBN 9783868991895



jQuery-Magneten

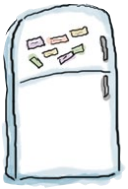
Bringen Sie die Codemagneten in die korrekte Reihenfolge, so dass der `zurueecksetzen`-Button wie gewünscht funktioniert. Ein paar Magneten haben wir schon für Sie platziert.

```
$("button#zurueecksetzen").click(function(){
    if (v == true){
        v = false;
    }
});
```

meine_skripts.js

```
"<li class="hamburger">Hamburger</li>");
$m[i]);
.before($f);
$(".portobello").replaceWith(
});
$(".tofu").remove();
$(".vorspeisen li").first()
$(this).after(
$(".tofu").each( function(i){
```

Dies ist ein Auszug aus dem Buch "jQuery von Kopf bis Fuß", ISBN 978-3-86899-189-5
http://www.oreilly.de/catalog/hfjqueryger/
Dieser Auszug unterliegt dem Urheberrecht. © O'Reilly Verlag 2011



jQuery-Magneten: Lösung

Bringen Sie die Codemagneten in die korrekte Reihenfolge, so dass der `zuruecksetzen`-Button wie gewünscht funktioniert. Ein paar Magneten haben wir schon für Sie platziert.

```
$("button#zuruecksetzen").click(function(){
    if (v == true){
        $(".portobello").replaceWith(    "<li class="hamburger">Hamburger</li>");
        $(".vorspeisen li").first()    .before($f);
        $(".tofu").each( function(i){
            $(this).after(    $m[i]);
        });
        $(".tofu").remove();
        v = false;
    }
});
```

Wir können die >>fleisch<<-Elemente wieder an ihre alte Position schieben, indem wir das $m-Array und den Index des >>tofu<<-Elements referenzieren, mit dem die Funktion gerade arbeitet.

meine_skripts.js

Dies ist ein Auszug aus dem Buch "jQuery von Kopf bis Fuß", ISBN 978-3-86899-189-5
http://www.oreilly.de/catalog/hfjqueryger/
Dieser Auszug unterliegt dem Urheberrecht. © O'Reilly Verlag 2011

Ryan Benedetti / Ronan Cranley, jQuery von Kopf bis Fuß, O´Reilly, ISBN 9783868991895

Das war's ... oder?

Alle Anforderungen für den »Zurücksetzen«-Button sind erfüllt. Wir müssen also nur noch unsere Projektdateien aktualisieren und sind fertig.

- ☑ Fügen Sie die Fisch-Vorspeisen wieder an ihrer ursprünglichen Position im DOM ein (also vor dem ersten Listeneintrag in der linken Spalte).
- ☑ Finden Sie Vorspeisen mit Portobello-Champignons und ersetzen Sie sie durch Hamburger.
- ☑ Finden Sie die Vorspeisen, die Tofu enthalten, und ersetzen Sie sie (in der richten Reihenfolge) durch die verschiedenen Fleischsorten.

Augenblick! Wir haben das P. S. auf der Serviette vergessen.

P. S.: Es wäre schön, wenn außerdem ein Blatt-Symbol neben den vegetarischen Alternativgerichten angezeigt werden könnte.

Hoppla! Ja, stimmt.

Zum Glück hat der Webdesigner in der CSS-Datei *meine_stile.css* bereits eine passende Klasse mit dem Namen `blattsymbol` angelegt.

```
.blattsymbol{
      list-style-image:url(../bilder/blatt.png');
      }
```

meine_stile.css

Schreiben Sie die Anweisung, die das Elternelement des Elternelements der »tofu«-Klassen mit dem Blatt-Symbol versieht.

.....

Tipp: addClass ist Ihr Freund.

Dies ist ein Auszug aus dem Buch "jQuery von Kopf bis
http://www.oreilly.de/catalog/hfjqueryger/
Dieser Auszug unterliegt dem Urheberrecht. © O'Reilly V
-86899-189-5

Lösung zur Übung

Nur noch ein bisschen im DOM herumklettern und etwas `addClass`-Zauberei, und wir sind fertig!

```
$(".tofu").parent().parent().addClass("blattsymbol");
```

Es gibt keine Dummen Fragen

F: Mit den anderen filter-Methoden komme ich klar, aber `slice` bringt mich immer noch wahnsinng durcheinander. Können Sie mir das noch einmal genauer erklären?

A: Die `slice`-Methode ist nicht leicht zu verstehen. Das Verwirrendste an `slice` sind seine Parameter: `slice(Start, Ende)`.
Der erste Parameter ist der `Start`-Parameter. Er muss angegeben werden, damit `slice` überhaupt funktioniert. Er gibt an, wo die Auswahl der Elemente im Array beginnen soll. Vergessen Sie nicht, dass das erste Element eines Arrays den Index »0« hat. Als `Start`-Parameter können Sie auch eine negative Zahl angeben. In diesem Fall wird vom Ende des Arrays an rückwärts gezählt und nicht vorwärts von seinem Anfang an.

F: Okay, und was macht dann der `End`-Parameter der `slice`-Methode?

A: Der zweite Parameter (Ende) muss nicht unbedingt angegeben werden. Wenn Sie ihn nicht angeben, wählt `slice` alle Elemente ab dem `Start`-Parameter und wählt von dort aus alle Elemente im Array aus, deren Index größer ist als der `Start`-Parameter. Der `End`-Parameter ist nicht unbedingt intuitiv, wenn Sie vergessen, dass die Zählung der Array-Elemente bei 0 beginnt.

F: Die `each`-Methode scheint ziemlich mächtig zu sein. Woher weiß `each`, mit welchem Element es gerade arbeitet?

A: Seine wahre Stärke entfaltet `each` erst, wenn Sie die Methode mit dem Schlüsselwort `this` kombinieren. Außerdem merkt sich `each` automatisch den Index des Elements, mit dem sie gerade arbeitet. Sie sollten nur mit `each` arbeiten, wenn Sie mehr als ein Element ausgewählt haben. Um das aktuelle Element zu referenzieren, können Sie `this` verwenden. Allerdings müssen Sie es mit dem jQuery-Kürzel umgeben: `$(this)`.

F: Warum muss ich in der Funktion `each` "i" oder "index" benutzen?

A: Die Indexvariable, oft "`i`" oder "`index`" genannt, wird von der `each`-Methode verwendet, um die bearbeiteten Elemente zu zählen. Auf diese Weise weiß `each`, wann sie mit ihrer Arbeit fertig ist. Ohne die Indexvariable wüsste `each` nicht, mit welcher Funktion sie arbeiten und wann sie mit der Arbeit aufhören soll.

F: Wie kann ich bestimmte Elemente in einem jQuery-Array finden?

A: Für die Suche nach Elementen in einem jQuery-Array gibt es die Methode `find`. Angenommen, Ihr Array enthält eine Reihe von li-Elementen:

```
var $meine_elemente =
$("li");
```

Um alle a-Elemente (Links) im Array zu finden, können Sie Folgendes schreiben:

```
$meine_elemente.find("a");
```

F: Gibt es in jQuery eine Möglichkeit, ein Element mit einem anderen zu umgeben?

A: Aber sicher. Um beispielweise ein Bild mit der ID `oreilly` mit einem Link zu umgeben, können Sie diesen Code hier verwenden:

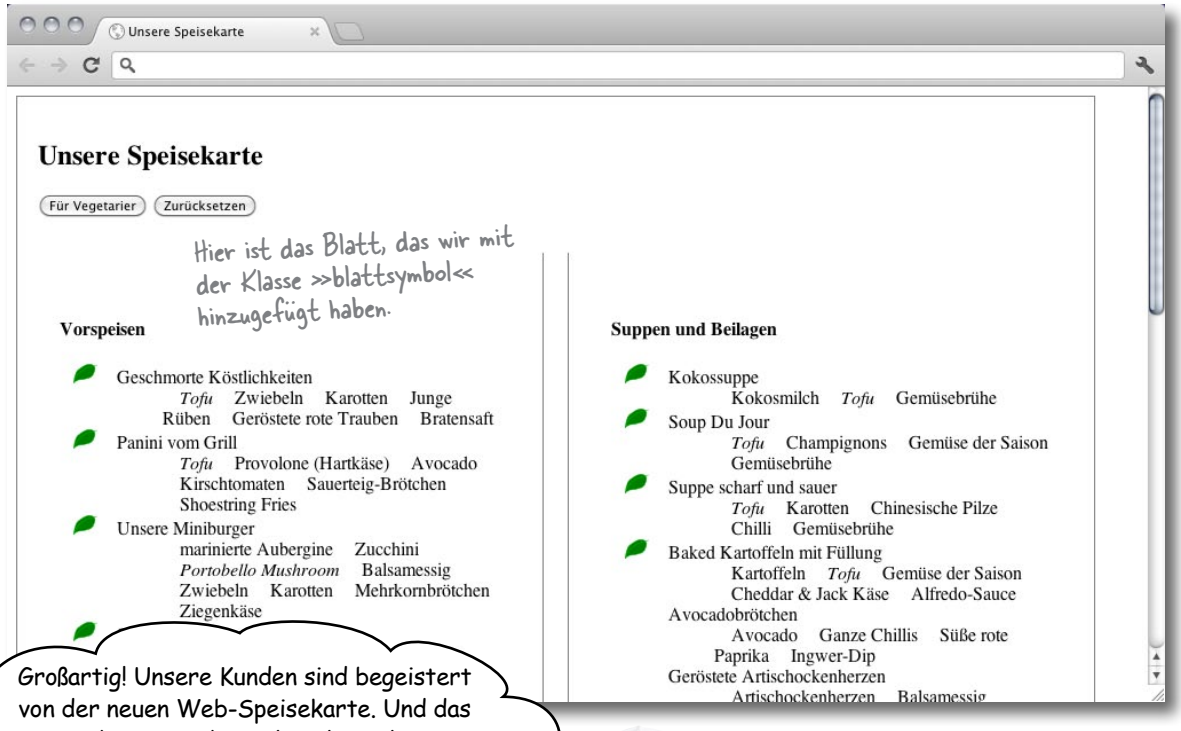
```
$("img#oreilly").wrap("<a
href='http://www.oreilly.
de'></a>");
```

Dies ist ein Auszug aus dem Buch "jQuery von Kopf bis Fuß", ISBN 978-3-86899-189-5
http://www.oreilly.de/catalog/hfjqueryger/
Dieser Auszug unterliegt dem Urheberrecht. © O'Reilly Verlag 2011

Ryan Benedetti / Ronan Cranley, jQuery von Kopf bis Fuß, O´Reilly, ISBN 9783868991895

PROBEFAHRT

Seit der letzen Aktualisierung Ihrer Dateien ist schon etwas Zeit vergangen. Erweitern Sie Ihr Skript um den Code für den »Zurücksetzen«-Button und den Code zum Hinzufügen und Entfernen der Klasse »blattsymbol«, um die vegetarischen Alternativen zu kennzeichnen. Zum Vergleich mit Ihrem Code finden Sie die Dateien für dieses Kapitel auch unter *http://www.thinkjquery.com/chapter04/*. Den eingedeutschten Code finden Sie unter *www.oreilly.de/catalog/hfjqueryger*.



Dies ist ein Auszug aus dem Buch "jQuery von Kopf bis Fuß", ISBN 978-3-86899-189-5
http://www.oreilly.de/catalog/hfjqueryger/
Dieser Auszug unterliegt dem Urheberrecht. © O'Reilly Verlag 2011

Ihr jQuery-Werkzeugkasten

Damit haben Sie Kapitel 4 abgeschlossen. Zusätzlich zu den bereits vorhandenen Dingen enthält der Werkzeugkasten jetzt die Möglichkeit, sich im DOM zu bewegen und etwas zu verändern, sowie Arrays und Filter.

DOM-Manipulation

Mit diesen Methoden können Sie nach Bedarf etwas zum DOM hinzufügen, ersetzen und entfernen:

- detach
- remove
- replaceWith
- before
- after

DOM-Traversal

Hier geht es darum, im DOM-Baum herumzuklettern und etwas zu verändern.

Um an die gewünschten Dinge zu gelangen, verwenden Sie die Beziehungen der Elemente zueinander und die dazugehörigen Methoden wie parent und child.

Das Verketten von Methoden ist eine effiziente Möglichkeit, sich schnell im DOM zu bewegen.

Arrays

Mit jQuery-Arrays können Sie alles Mögliche speichern, auch Elemente, um es später wiederzuverwenden.

Wie bei Variablen stellen Sie Ihrem Array ein $ voran, um anzuzeigen, dass hier spezielle jQuery-Sachen gespeichert werden.

Filter

Filtermethoden helfen dabei, die ausgewählten Elemente auf eine Untermenge zu beschränken:

- first
- equal
- last
- slice
- filter
- not

Dies ist ein Auszug aus dem Buch "jQuery von Kopf bis Fuß", ISBN 978-3-86899-189-5
http://www.oreilly.de/catalog/hfjqueryger/
Dieser Auszug unterliegt dem Urheberrecht. © O'Reilly Verlag 2011

Ryan Benedetti / Ronan Cranley, jQuery von Kopf bis Fuß, O´Reilly, ISBN 9783868991895