

Autocomplete: script.aculo.us

== Was ist Autocomplete ==

Die getippten Buchstaben des Suchwortes werden unmittelbar im Hintergrund zum Server übertragen, das jeweilige Ergebnis zurückgeschickt und "live" in die bestehende Seite eingefügt. Dieser Vorgang wiederholt sich bei jedem getippten Buchstaben. Der Service wurde besonders populär nachdem Google ihn lange in der Beta Phase hatte und inzwischen auf google.com produktiv einsetzt:

easy-coding.de/Attachment/516/...ecd2fe764b7e5b384492484a2

== Was ist script.aculo.us ==

script.aculo.us ist eine JavaScript-Bibliothek, die auf dem JavaScript-Framework Prototype aufbaut. Sie bietet dynamische visuelle Effekte und GUI-Elemente über das Document Object Model (DOM).

Warum sollte man script.aculo.us den vielen anderen JavaScript Frameworks vorziehen? Das soll an dieser Stelle nicht diskutiert werden. Ich werde den Autocomplete Effekt mit mehreren Frameworks erklären.

Links:

- [Offizielle Seite des Frameworks](#)
- [Dokumentation für die Autocompleter Schnittstelle](#)

== Ladegrafik ==

Als Vorbereitung müsst ihr euch eine Grafik beschaffen. Ihr könnt die Grafik entweder mit GIMP, Photoshop & Co selbst erstellen oder aber den kostenlosen Webservice von ajaxload.info nutzen. Ihr braucht für dieses Tutorial keine Kenntnisse aus dem [HowTo \[wiki\]AJAX Ladegrafik\[/wiki\]](#), da script.aculo.us seine eigenen Helperfunktionen mitbringt um die Funktionalität einer Ladegrafik zu erzeugen.

== Code ==

=== Frontend ===

script.aculo.us ist wie viele andere JavaScript Frameworks sehr "ID-bezogen". Ihr werdet feststellen, dass ihr die IDs der Container an anderen Stellen im JavaScript Code angebt.

Quellcode

```
1. <!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.1//EN" "http://www.w3.org/TR/xhtml11/DTD/xhtml11.dtd">
2. <html xmlns="http://www.w3.org/1999/xhtml" dir="ltr" xml:lang="de">
3. <head>
4. <meta http-equiv="content-type" content="text/html; charset=UTF-8" />
5. <title>Autocomplete: script.aculo.us</title>
6. <script src="js/prototype.js" type="text/javascript"></script>
7. <script src="js/scriptaculous.js?load=effects,controls" type="text/javascript"></script>
8. <link rel="stylesheet" type="text/css" href="style.css" />
9. </head>
10. <body>
11. <h2>Autocomplete: script.aculo.us</h2>
12. <form action="" method="post">
13. <div>
14. Suchen nach:
15. <input id="sourceInput" type="text" name="sourceInput" />
16. 
17. <div id="destinationDiv" class="autocomplete"></div>
18. <input type="submit" value="Speichern" />
19. </div>
20. </form>
21. <script type="text/javascript">
22. new Ajax.Autocompleter('sourceInput','destinationDiv','search.php', {
23. indicator: 'loadingImg'</pre></div>
```

```

27. });
28. //]]></script>
29. </body>
30. </html>

```

Alles anzeigen

Um die Serverlast zu verringern bietet script.aculo.us zwei Parameter. Der Parameter **frequency** gibt die Zeitspanne in Sekunden an die mindestens zwischen zwei AJAX Anfragen vergehen muss. **minChars** kann man erhöhen um zu erreichen, dass die AJAX Anfragen erst ab einer bestimmten Mindest-Zeichenlänge gesendet werden.

Die Parameter ändert man beim konstruieren des Autocompleters - die Standardwerte sind 0.4 und 1.

Quellcode

```

1. new Ajax.Autocompleter('sourceInput','destinationDiv','search.php', {
2. indicator: 'loadingImg',
3. frequency: 0.4,
4. minChars: 1
5. });

```

=== Backend ===

Das Backend muss einen normalen HTML Code ausgeben. am besten ist eine Darstellungsvariante unterhalb des Eingabeelements. Außerdem kann man die Liste scrollbar machen. Das erreicht man durch die Angabe einer festen Größe und der Eigenschaft overflow:auto.

Wenn ihr Hilfe zum Umgang mit dem PDO braucht, könnt ihr das im entsprechenden Tutorial nachlesen: [\[wiki\]Einführung in PDO\[/wiki\]](#) - generell könnt ihr aber auch jede andere Datenbankschnittstelle dazu benutzen.

Quellcode

```

1. <?php
2. require_once 'Configuration.php';
3. require_once 'MyDB.php';
4. $sql = "SELECT username,
5. job
6. FROM user
7. WHERE username LIKE :username
8. LIMIT 10";
9. $stmt = MyDB::getInstance()->prepare($sql);
10. $stmt->execute(array(
11. ':username' => "%".$_REQUEST['sourceInput']."%",
12. ));
13. $result = $stmt->fetchAll(PDO::FETCH_ASSOC);
14. echo '<ul style="overflow:auto;height:150px">';
15. foreach($result as $row) {
16. printf('<li><b>%s</b> %s</li>', $row['username'], $row['job']);
17. }
18. echo '</ul>';
19. ?>

```

Alles anzeigen

== Demo ==

Ihr könnt den zusammengesetzten Code hier live ausprobieren: demo.easy-coding.de/ajax/autocomplete-script.aculo.us/.
Außerdem gibt es alle Dateien [zum kostenlosen Download](#)