

jQuery – Grundlagen

Ingo Köster

Diplom-Informatiker (FH)

Was ist jQuery?

- › Ein OOP-Framework für JavaScript, ursprünglich von John Resig
- › Open Source, vertrieben unter der MIT- und GPL-Lizenz
- › Cross-Browser
- › Klein und kompakt (wenige kB)
- › Wird von sehr vielen Web-Seiten eingesetzt
- › Große Entwicklergemeinschaft, starke Unterstützung durch z.B. Google, MS

Was ist jQuery?

- › Elegante und mächtige Syntax
- › Erlaubt es schnell moderne Web-Features einzusetzen:
 - › AJAX
 - › Effekte und Animationen
- › Schnell zu erlernen

Quellen

- › jquery.com - Hauptseite mit der neusten Version
- › api.jquery.com - Dokumentation
- › plugins.jquery.com - Viele nützliche Erweiterungen
- › jqueryui.com - UI-Widgets und weitere Effekte

jQuery Versionen

- › Seit 2006 jQuery Version 1.x
 - › Wird weiterhin gewartet und aktualisiert (keine neuen Features mehr)
- › Ab 2013 jQuery 2.x
 - › Keine Unterstützung für Internet Explorer 6, 7 und 8
 - › Wird weiterhin gewartet und aktualisiert (keine neuen Features mehr)
- › Ab 2016 jQuery 3.x
 - › Enthält einem Slim-Build, welcher auf einige Module verzichtet (ajax und effects)

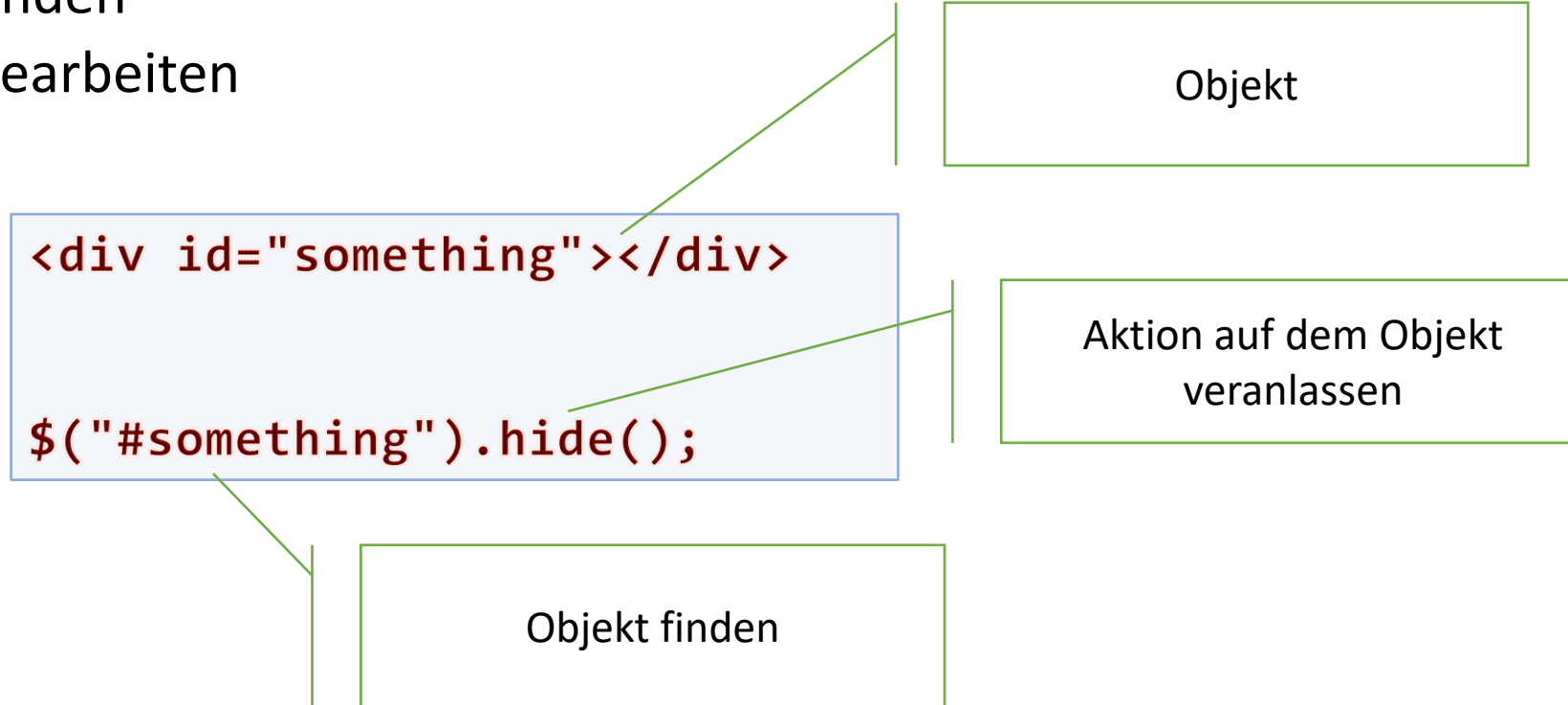
Komprimiert vs. Unkomprimiert

- › Die jQuery Bibliothek liegt komprimiert und unkomprimiert vor
- › Komprimiert für Praxis-Einsatz (`.min` im Namen)
 - › Keine Kommentare
 - › So wenig Zeichen wie möglich im Code:
 - ›

```
(function(a,b){function b$(a){return d.isWindow(a)?a:a.nodeType===9?a.defaultView||a.parentWindow:!1}function bX(a){if(!bR[a]){var b=d("<"+a+">").appendTo("body")...
```
- › Unkomprimiert für den Einsatz während der Entwicklung

Das jQuery-Prinzip

- › Bevor man mit jQuery anfängt, sollte man sich das Grundprinzip merken und dieses dann stets anwenden!
 - › Objekt finden
 - › Objekt bearbeiten



jQuery Beispiele

› Beispiele: Elemente verstecken

- › `$(this).hide();` Das aktuelle Element
- › `$("p").hide();` Alle `<p>` Elemente
- › `$(".test").hide();` Alle Elemente mit `class="test"`
- › `$("#test").hide();` Das Element mit `id="test"`
- › `$("#test div").hide();` Alle `<div>` Elemente unterhalb von `id="test"`

› Verwendet CSS Selektoren (Sizzle Selector Engine)

- › http://www.w3schools.com/cssref/css_selectors.asp

jQuery einbinden

- › Herunterladen und zu den Daten der Webseite legen

```
<head>  
<script src="jquery-1.9.1.min.js"></script>  
</head>
```

- › Alternative: Nicht selbst hosten, sondern jeden Client jQuery von einem CDN (Content Delivery Network) beziehen lassen

```
<head>  
<script  
  src="//ajax.googleapis.com/ajax/libs/jquery/1.9.1/j  
  query.min.js">  
</script>  
</head>
```

jQuery einbinden - Beispiel

```
<html>
<head>
  <title>jQuery Demo</title>
  <script src="jquery-1.9.1.min.js"></script>
  <script type="text/javascript">
    $(document).ready(function() {
      $("body")
        .hide()
        .html("<h1>Hello, jQuery!</h1>")
        .fadeIn(2500);
    });
  </script>
</head>
<body>
  <!-- Die Seite ist leer ... -->
</body>
</html>
```

jQuery

- › Kern von jQuery ist die `$()` Funktion
 - › Alternativ mittels `jQuery()`
- › Sogenannte Factory Funktion
- › Der Übergabeparameter bestimmt womit gearbeitet wird
- › jQuery stellt nicht nur Zugriff auf das gewählt Objekt bereit, sondern erweitert dieses um neue Fähigkeiten
 - › Sogenannter Wrapper
- › D.h. der Aufruf von `$()` liefert immer jQuery-Objekt zurück!

jQuery - Knotenliste

- › Das jQuery-Objekt kann auch eine Ansammlung von z.B. `<p>` Elementen sein (Knotenliste)
- › jQuery verwendet alle Objekte der Liste und wendet eine angehängt Methode auf jedes Objekt an
- › Beispiel: An alle gefundenen `<p>` Elemente die CSS-Klasse `wichtig` anhängen
 - › `$("p").addClass("wichtig");`
 - › Implizite Schleife
- › Mittels `length` die Anzahl der Elemente ermitteln
 - › `$("p").length`

Ergebnismengen durchlaufen

- › Jedes zweite <p> in Grün einfärben

```
$("#p").each(function(i)
{
    if (i % 2 === 0)
    {
        $(this).css("color", "green");
    }
});
```

jQuery - Kompatibilität

- › Bei Konflikten wegen der `$()` Funktion kann jQuery auch eine andere Variable verwenden
 - › Z.B. durch Verwendung von anderen Frameworks
 - › `$(...).noConflict()`

- › Beispiel:

```
let $j = jQuery.noConflict();  
$j(document).ready(function () {  
    $j("#something").hide();  
});
```

Verkettung - Chaining

- › Methodenverkettung
- › Die meisten jQuery-Methoden liefern ein Objekt zurück, für welches wieder eine Methode aufgerufen werden kann usw.
- › Erst Ausblenden, dann wieder einblenden
 - › `$("#ziel").fadeOut().fadeIn();`
- › Ablauf:
 - › `let x = $("#ziel");`
 - › `let y = x.fadeOut();`
 - › `y.fadeIn();`

jQuery - Utilities

- › `$(...).each()`
- › `$(...).grep()`
- › `$(...).data()`
- › `$(...).map()`
- › `$(...).merge()`
- › `$(...).parseJSON()`
- › `$(...).parseXML()`

jQuery – Events

› Etwas ausführen wenn das Dokument geladen ist

```
$(document).ready(function() {
```

```
    ...
```

```
});
```

jQuery – DOM-Manipulation

- › `$(...).append()`
- › `$(...).appendTo()`
- › `$(...).attr()`
- › `$(...).css()`
- › `$(...).after()`
- › `$(...).html()`
- › `$(...).text()`

jQuery - Animationen

- › `$(...).show()`
- › `$(...).hide()`
- › `$(...).slideDown()`
- › `$(...).slideUp()`
- › `$(...).fadeIn()`
- › `$(...).fadeOut()`

Attribute

- › Nach dem Attribut `target` suchen
 - › `$("[target]")`
- › Attribut setzen mittels `.attr()`
- › Alle Links werden in einem neuen Fenster bzw. Tab geladen
 - › `$("a").attr("target", "_blank");`

CSS

- › CSS-Eigenschaften einem (oder mehreren) Element(en) hinzufügen
- › Rahmen eines `<div>` der Klasse "note" festlegen bzw. verändern
 - › `$("#div.note").css("border", "2px solid black");`