



JavaScript und JavaScript Frameworks

CSS Eigenschaften und Attribute



Übersicht und Zielstellung

- In diesem Abschnitt erfahren Sie:
 - wie die Existenz von CSS-Klassen geprüft wird,
 - wie man CSS-Klassen setzen und entfernen kann,
 - wie CSS-Style Elementen zugewiesen werden,
 - wie man Abmessungen und Positionen ermittelt,
 - wie man das Scrollen auswerten und steuern kann,



Start in das Thema

CSS-Eigenschaften und Klassen

Einführende Übersicht

Mit jQuery CSS-Eigenschaften und -Klassen setzen

jQuery bietet verschiedene Methoden zur Bearbeitung des Styles von Elementen an. Diese Methoden greifen auf das **class-Attribut** oder direkt auf das **style-Property eines Knotens** zu.

Um CSS-Eigenschaften direkt zu setzen, gibt es die allgemeine Methode **.css()**. Für die Breite oder Höhe von Containern können auch die Shortcuts **.width()** und **.height()** verwendet werden.

Um Klassen hinzuzufügen, zu entfernen oder umzuschalten, gibt es die Methoden **.addClass()**, **.removeClass()** oder **.toggleClass()**.

Mit jQuery CSS-Eigenschaften und -Klassen lesen oder prüfen

Manchmal möchte man nicht gleich die CSS-Eigenschaften ändern, sondern vorher ihren Wert herausfinden. jQuery löst dieses Problem und ermöglicht das Lesen auch in diesen Fällen.

Die Methode **.css()** steht für gängige Eigenschaften zur Verfügung.

Für Spezialfälle wie Abmessungen oder Positionen von Containern gibt es eine Reihe von Methoden wie **.innerHeight()**, **.innerWidth()**, **.outerHeight()**, **.outerWidth()** und weitere.

Um zu prüfen, ob ein Element eine bestimmte CSS-Klasse hat, kann die Methode **.hasClass()** verwendet werden.

Das Attribut class

Methoden für das Attribut class

Eine typische Situation besteht darin, festzustellen, ob eine bestimmte Klasse einem Element zugewiesen ist.

Die Methode **.hasClass()** ermöglicht es, den Selektor zu prüfen, ob bereits eine Klasse zugewiesen ist. Ist die geprüfte Klasse vorhanden, gibt die Methode ein **true** zurück.

Beispiel HTML-Code: `<div id="demo" class="container"> ... </div>`

jQuery Code: `$('#demo').hasClass('container');`

Das Ergebnis: Der Rückgabewert ist true. Dies gilt auch, wenn mehrere Klassen zugewiesen sind.

`<div id="demo" class="container test"> ... </div>`

Die Prüfung des class-Attributs erfolgt, ob es den übergebenen Prüfstring als Teilstring enthält.

Hinweis: der gesamte Ausdruck durch **.hasClass()** ergibt einen booleschen Wert ergibt.

Es entsteht kein jQuery-Objekt. An die Methode **.hasClass()** kann keine weitere jQuery-Methode angereiht werden kann!

Das Attribut class

Probleme mit der Methode .hasClass()

Sobald mehrere Klassen gleichzeitig auf ihre Existenz getestet werden sollen, gibt es mit der Methode **.hasClass()** Schwierigkeiten. Es können zwar mehrere Klassenbezeichner getrennt durch Leerzeichen angegeben werden, allerdings wird dann der komplette String verglichen.

Der HTML Code `<ul class="class01 class02">`

Der jQuery Code `$("ul").hasClass("class01 class02");`

Hier wäre das Ergebnis **true**, da der String identisch ist.

aber: `$("ul").hasClass("class02 class01");`

Hier wäre das Ergebnis **false**, der String ist nicht identisch.

Eventuell:
`$("ul").hasClass("class01");`
`$("ul").hasClass("class02");`

Tipp: Das Prüfen von zwei oder mehr Klassen mit **.is()** ist für diese Prüfung besser geeignet.

Eine Methode, die einen **CSS-Selektor** anstatt eines Klassennamens akzeptiert

```
$("ul").is(".class02.class01") // gibt true zurück
```

Arbeiten mit CSS-Klassen

Hinzufügen und Entfernen von CSS-Klassen

Häufig angewandte Aktionen in Zusammenhang mit CSS-Klassen sind das Hinzufügen bzw. Entfernen von Klassen

Webseiten und Präsentationen können sehr schnell geändert werden

CSS3-Transitions und CSS3-Animationen können ganz einfach über Klassen gesteuert werden

Die Methode .addClass()

Mit der Methode .add-Class() kann einem Element eine Klasse hinzugefügt werden.

Definieren wir im <style>-Element ein Klasse.

```
.container {  
    width:300px;  
    height:150px;  
    background-color:#CCCCCC;  
}
```

Folgendes Element soll im <body> stehen.

```
<div> Hier kann ein beliebiger Inhalt stehen </div>
```

Das jQuery-Skript:

```
$(document).ready(function() {  
    $("div").addClass("container");  
})
```

Arbeiten mit Klassen

Die Methode `.addClass()` bietet die Möglichkeit eine anonyme Funktion aufzurufen. Diese Funktion erhält zwei Parameter.

Der erste ist der Index des aktuell verarbeiteten Elements der Collection, der zweite Parameter ist der Wert des class-Attributs

Eine unsortierte Liste an, deren zweites Element die Klasse `auswahl` besitzt:

```
<ul>
  <li>Thema 1</li>
  <li class="auswahl">Thema 2</li>
  <li>Thema 3</li>
</ul>
```

In der Methode `.addClass()` rufen wir eine Funktion auf und prüfen in deren Codeblock, ob die Klasse `auswahl` anliegt

```
$(document).ready(function() {
  $("li").addClass(function(index, class) {
    // liegt "auswahl" nicht an ...
    if (class != auswahl) {
      // ... dann folgende Klasse zuweisen:
      return "li-element-" + index;
    }
  });
});
```

Ergebnis:

```
<ul>
  <li class="li-element-0">Thema 1</li>
  <li class="auswahl">Thema 2</li>
  <li class="li-element-2">Thema 3</li>
</ul>
```

Sie können damit jedem der einzelnen Listenelemente eine eigene Klasse zuweisen.

Kleine Übung

Erstellen Sie ein kleines Beispiel nach folgenden Vorgaben:

1. Aufgabe: Legen Sie ein neues Beispiel Dokument an. `jQueryUebung10.html`
2. Aufgabe: Legen Sie eine Überschrift an!
Erstellen Sie eine ungeordnete Liste mit vier Listen-Elementen.
3. Aufgabe: Tragen Sie beliebige Werte/Texte in die vorhandenen Elemente ein
4. Aufgabe: Legen Sie eine Klasse im Style an, die einen Rahmen enthält, Abstände nach innen hat und eine geeignete Hintergrundfarbe hat.
5. Aufgabe: Weisen Sie drei von den Listen-Elementen die Klasse zu.

Arbeiten mit Klassen

Die Methode `.removeClass()` bietet die Möglichkeit eine oder mehrere Klassen zu entfernen.

Mit Angabe des Klassenbezeichners können Klassen direkt aus der Collection entfernt werden:

```
$ ("div").removeClass ("container");
```

Bei Einsatz einer Liste können mehrere Klassen aus der Collection entfernt werden:

```
$ ("div").removeClass ("container", "auswahl");
```

Mit der „Breachstange“ können alle Klassen aus der Collection entfernt werden, dazu wird weder eine Klasse oder Liste angegeben:

```
$ ("div").removeClass ();
```

Bei einer Neuorganisation von Klassenzuweisungen kann man auch erst alle Klassen entfernen und anschließend die „Wunschklasse“ hinzufügen:

```
$ ("div").removeClass ().addClass ("wunschklasse");
```

Anwendung der Methode .attr()

Alternative Methode zum Ändern des Attributes class

Um alle Klassen zu entfernen und neue zu setzen hatten wir:

```
$("div").removeClass().addClass("wunschklasse");
```

Mit der Methode .attr("Attributname", "Wert des Attributes") wird ein Attribut neu gesetzt und mit dem Wert versehen.

Ist das Attribut bereits vorhanden, wird es komplett überschrieben.

```
$("div").attr("class", "wunschklasse");
```

Damit werden alle Klassen entfernt.

Um dieses zu vermeiden, nutzen wir die beschriebenen Methoden `.addClass(class)` und `.removeClass(class)`.

Arbeiten mit Klassen

Die Methode .toggleClass()

Wenn an einem Element eine Klasse mal anliegen soll und mal nicht – ein Hin- und Herwechseln – bietet sich die Methode `.toggleClass()` an.

Die Methode überprüft im Hintergrund den Zustand der Collection und fügt dann die Klasse hinzu oder entfernt sie.

Für eine Demo erstellen wir folgenden HTML-Code.

```
<p>Klick mich.</p>
<p>Und nun klick mich auch.</p>
<p>Und mich auch!</p>
```

Dann benötigen wir zwei Style-Definitionen.

```
<style type="text/css">
  .auswahl { background-color:#CCCCCC; }
  p { margin:10px 50px 0 10px;
      padding:5px;
      cursor:pointer; }
</style>
```

Der einfache Umschalter per jQuery-Code.

```
$(document).ready(function() {
  $("p").click(function () {
    $(this).toggleClass("auswahl");
  });
});
```


Arbeiten mit Klassen

Die Methode `.toggleClass()` mit einem Zähler

Um die Anzahl der Klicks für eine Reaktion zu nutzen, kann in der Methode um einen Parameter erweitert werden.

```
$(document).ready(function() {  
    // externer Zähler:  
    var i = 0;  
    $("p").click(function () {  
        // Zähler iterieren:  
        i++;  
        $(this).toggleClass("auswahl", (i % 3 == 0));  
        $(this).html(" Jeder dritte Klick: " + (i % 3 == 0))  
    });  
});
```

Des Weiteren bietet Ihnen die Methode `.toggleClass()` die Möglichkeit, eine anonyme Funktion zu übergeben.

```
$("p").click(function() {  
    $(this).toggleClass(function() {  
        return "classname-" + $(this).index();  
    });  
});
```

Kleine Übung

Erstellen Sie ein kleines Beispiel nach folgenden Vorgaben:

1. Aufgabe: Legen Sie ein neues Beispiel Dokument an. `jQueryUebung11.html`
2. Aufgabe: Legen Sie eine Überschrift an!
Erstellen Sie vier Absätze.
3. Aufgabe: Tragen Sie beliebige Werte/Texte in die vorhandenen Elemente ein
4. Aufgabe: Legen Sie eine Klasse im Style an, die einen Rahmen enthält, Abstände nach innen hat und eine geeignete Hintergrundfarbe hat.
5. Aufgabe: Erstellen Sie den jQuery-Code, der den Absätzen auf Mausklick die Klasse zuweist oder sie wieder entfernt.

Arbeit mit CSS Style Definitionen

Die Methode .css()

Mit der Methode kann das style-Attribut eines Knotens im DOM beeinflussen.

Die Methode kann beliebige gesetzte CSS-Eigenschaften eines Elementes auslesen.

Die Methode kann beliebige CSS-Eigenschaften in ein Elementes schreiben.

Wichtig! Beim Lesen wird immer nur das erste Element erfasst, beim Schreiben jedoch alle Elemente.

Die Methode **.css()** bietet verschiedene Möglichkeiten, die durch die übergebenen Parameter individuell ausführbar macht.

Zum Lesen einer Eigenschaft muss diese als String übergeben werden:

```
$("#div").click(function () {  
    // Hintergrundfarbe auslesen:  
    var farbe = $(this).css("background-color");  
    alert("Dieser Div hat die Hintergrundfarbe "+ color);  
});
```

Beim Setzen einer CSS-Eigenschaft, wird als zweiter Parameter der gewünschte Wert als String übergeben

```
$("#div").click(function () {  
    // Schriftfarbe auf Orange setzen:  
    $(this).css("color", "#ff9900");  
});
```

Arbeit mit CSS Style Definitionen

Die Methode .css()

Der zu setzende neue Wert kann auch aus dem Ergebnis einer Abfrage entnommen werden.

Beispielsweise können Sie die Schriftfarbe eines Containers mit der Schriftfarbe eines anderen steuern, indem Sie dessen Farbwert auslesen und übergeben:

```
$("#div").click(function () {  
    // Schriftfarbe auf Schriftfarbe von #p1 setzen:  
    $(this).css("color", function(){  
        // Schriftfarbe von #p1 auslesen und zurückgeben:  
        return $("#p1").css("color");  
    });  
});
```

Es können auch mehrere Werte gleichzeitig gesetzt werden. Die Auflistung wird im jQuery als Map bezeichnet.

```
$("#div").click(function () {  
    // Drei Eigenschaften über ein Objekt setzen:  
    $(this).css( {"color":"#ff9900", "background-color":"black", "font-weight":"bold"} );  
});
```


Arbeit mit CSS Style Definitionen

Methoden zum Setzen *einer* Eigenschaft mit .css() - Caveats

Es gibt zwei Wege, mit der Methode .css() eine einzelne Eigenschaft zu setzen..

Anwendung von zwei Parametern (Bezeichner und Wert)

```
// zwei Argumente:  
$(this).css( "color", "#ff9900" );
```

Übergabe *eines* Objektes

```
// ein Argument (Objekt):  
$(this).css( { color: "#ff9900" } );
```

Kleine Übung

Erstellen Sie ein kleines Beispiel nach folgenden Vorgaben:

1. Aufgabe: Legen Sie ein neues Beispiel Dokument an. `jQueryUebung12.html`
2. Aufgabe: Legen Sie eine Überschrift an!
Erstellen Sie eine Liste mit vier Elementen, der Text der Listen-Elemente soll sein
rot, grün, blau, grau
Erstellen Sie einen Absatz. Der Absatz kann einen einfachen Text enthalten
3. Aufgabe: Sorgen Sie nun dafür, dass auf Mausklick auf ein Listen-Element die Hintergrundfarbe
des Absatzes entsprechend einer `style="..."` angepasst wird.

Abmessungen und Positionen von Containern

Abmessungen von Containern

Die Abmessungen von Containern im Layout sind zum Teil schwer vorherzusagen und hängen von einer Vielzahl von Bedingungen ab.

Die meisten Eigenschaften werden vom Browser beim Rendern der Seite bestimmt (**computed height, computed width**) .

Auch hier bietet jQuery einige Methoden an, diese in manchen Browsern nicht greifbaren Größen auszulesen.

Je nach Einsatz werden bei den Ermittlungen, die eigentliche Höhe, die Paddings und Borders mit berücksichtigt

Der HTML-Code für das Beispiel:

```
<div id="box01">
  <p id="p01"></p>
  <p id="p02">Lorem ipsum ...</p>
  <p id="p03">Lorem ipsum ...</p>
  <p id="p04">Lorem ipsum ...</p>
</div>
```

Der jQuery-Code für das Beispiel:

```
$(document).ready(function() {
  $('#p01').text(
    'Box1 hat eine width() von ' + $('#box01').width() +
    'px und eine .innerWidth() von ' + $('#box01').innerWidth() +
    'px. Ihre Höhe height() beträgt ' + $('#box01').height() +
    'px, ihre innerHeight() ' + $('#box01').innerHeight() +
    'px. Die outerWidth() ist ' + $('#box01').outerWidth() +
    'px, die outerHeight() ' + $('#box01').outerHeight() + 'px.' );
});
```

Abmessungen und Positionen von Containern

Position von Containern

Auch die Position der Container, unabhängig ob diese positioniert oder gefloatet sind, wird erst beim Rendering der Seite durch den Browser bestimmt..

jQuery verfügt über Methoden, um die Koordinaten der Container zu ermitteln.

Folgender HTML-Quellcode dieses Beispiels besteht aus drei ineinander verschachtelten <div>-Containern, die zum Teil relativ, zum Teil absolut positioniert sind:

```
<div id="wrap">
  <div id="box1">Dies ist Box 1.
    <div id="box2">
      Dies ist Box 2 in div#box1.
    </div>
  </div>
</div>
```

```
div#wrap { margin:30px; padding:10px; height: 200px;
width:500px; border:1px solid #aaa; background-color:#eee;
}
div#box1 { position:relative; height: 120px; width:250px;
margin:20px; border:1px solid #aaa; background-color:#ddd;
}
div#box2 { position:absolute; top:65px; left:180px; height: 70px;
width:200px; border:1px solid #aaa; background-color:#bbb;
}
```


Abmessungen und Positionen von Containern

Die Methode .offset()

Die Methode gibt ein Objekt zurück, das in den Eigenschaften top und left die Koordinaten des übergebenen Objekts enthält.

Das betroffene Objekt muss sichtbar und im DOM-Baum eingehängt sein.

```
$(document).ready(function() {  
    $("#div#box2").click(function () {  
        var myOffset = $(this).offset();  
        alert('Box 2: offset() ist , +  
        myOffset.left + 'px left, , +  
        myOffset.top + 'px top. ');  
    });  
});
```

Das Ergebnis top:127px und left:250px.

Die Angaben der Positionierung sind nicht die Koordinaten gegenüber Box 1, sondern im Bezug auf den Dokumentursprung

Abmessungen und Positionen von Containern

Die Methode .offset()

Die Methode .offset() kann auch zum Positionieren genutzt werden.

Für die Positionierung der Box 2 an die Koordinaten top:150px und left:300px gilt folgender Code:

```
$(document).ready(function() {  
    $("#div#box2").click(function () {  
        $(this).offset( {top:150, left:300} );  
        var myOffset = $(this).offset();  
        alert('Box 2: offset() ist , +  
        myOffset.left + 'px left, , +  
        myOffset.top + 'px top. ');  
    });  
});
```

Auch hier beziehen sich die Koordinaten auf den Ursprung des Dokumentes

Abmessungen und Positionen von Containern

Man kann natürlich mit jQuery auch Positionen ermitteln, neue festlegen und dann neu zuweisen

Der jQuery-Code für ein einfaches Beispiel kann so aussehen

```
$(document).ready(function() {  
    $("#div#box2").click(function () {  
        var myOffset = $(this).offset();  
        $(this).offset({ top: myOffset.top+50, left: myOffset.left+50 });  
        var myOffset = $(this).offset();  
        alert('Box 2: offset() ist , + myOffset.left + 'px left, ' +myOffset.top + 'px top.');    });  
});
```

Bei jedem Mausklick auf das Element

- wird die Position ermittelt,
- anschließend wird das Element top: und left: jeweils um 50px verschoben,
- dann wird die neue Position ermittelt und ausgegeben

Abmessungen und Positionen von Containern

Methode `.offsetParent()`

Die Methode `.offsetParent()` gibt den Bezugscontainer, bei dem es sich um den nächsten positionierten Container nach oben innerhalb der Dokumenthierarchie handelt, als jQuery-Objekt zurück:

```
$(document).ready(function() {  
    $("div#box2").click(function () {  
        var myOffsetParent = $(this).offsetParent();  
        myOffsetParent.css({border: '2px dashed #555'});  
        alert('Box 2: offsetParent() ist ' + myOffsetParent.attr('id') + ' ');  
    });  
});
```


Kleine Übung

Erstellen Sie ein kleines Beispiel nach folgenden Vorgaben:

1. Aufgabe: Legen Sie ein neues Beispiel Dokument an. `jQueryUebung13.html`
2. Aufgabe: Legen Sie eine Überschrift an!
 Legen Sie ein `<div>`-Element an, in dem Sie eine Überschrift und einen Absatz einbinden
 Legen Sie ein weiteres `<div>`-Element an, in dem Sie eine Liste mit vier Elementen anlegen
 Alle Elemente sollen beliebigen Text enthalten
3. Aufgabe: Erstellen Sie ein jQuery-Code, der Ihnen die Positionen der beiden `<div>`-Elemente durch eine `alert()` Funktion ausgegeben werden soll.

Abmessungen und Positionen von Containern

Methode `.position()`

Die Methode `.position()` ähnelt der Methode `.offset()`, kann jedoch die Koordinaten nur auslesen und nicht verändern.

Wie auch bei `.offsetParent()` gibt `.position()` die Werte aus, mit denen `div#box2` laut CSS absolut zur `div#box1` positioniert wurde:

```
$(document).ready(function() {  
    $("#div#box2").click(function () {  
        var myPosition = $(this).position();  
        alert('Box 2: position() ist ' + myPosition.left + 'px left, ' + myPosition.top + 'px top.');    });  
});
```

Abmessungen und Positionen von Containern

Wenn der Viewport kleiner als das Layout ist, werden sich Bestandteile des Inhaltes außerhalb des Viewports befinden
Das kann entweder in horizontaler und/oder vertikaler Richtung sein.

Scrollen und die Ermittlung der Scroll-Position lassen sich auch per jQuery umsetzen.

Die Methoden **.scrollLeft()** und **.scrollTop()**.

Beide Methoden ermöglichen es, die Scroll-Position zu ermitteln bzw. die Position zu setzen.

Beachten Sie, dass **.scrollTop()** und **.scrollLeft()** nicht die Position eines Elements gegenüber dem Viewport auslesen, sondern die Scroll-Position innerhalb eines Elements.

Abmessungen und Positionen von Containern

Folgender HTML-Code soll dazu dienen, um das Scrollen des Dokuments auszuprobieren:

```
<div id="box1">
  <b>Scrollposition:</b>
  <div><!-- Ausgabe --></div>
  <a href="javascript:sprung()">Spring zu Box 2</a>
</div>
<div id="box2"></div>
```

Zwei <div>-Container, von denen der eine (div#box1) fixiert positioniert werden soll. Die Scrollposition soll darin auszugeben werden. Der zweite Container (div#box2) ist absolut außerhalb des Viewports positioniert werden, um das Scrollen auslösen zu können.

```
body {
  width:5000px;
  height:5000px;
}
```

```
#box1{
  position:fixed;
  top:10px;
  right:10px;
  ...
}
```

```
#box2{
  position:absolute;
  top:350px;
  left:1200px;
  ...
}
```

Abmessungen und Positionen von Containern

Mit der folgenden jQuery-Funktion wird die #box1 an die Position der #box2 gescrollt (hier eigentlich das Fenster-Objekt) :

```
function jump() {  
    // Position von Box 2 bestimmen:  
    var sTop = $('#box2').offset().top;  
    // zur ausgelesenen Position scrollen:  
    $(window).scrollTop(sTop);  
    $(window).scrollLeft(sLeft);  
}
```

Nun können wir noch die Ausgabe der aktuelle Position definieren

```
$(document).ready(function() {  
    $(window).scroll( function() {  
        $('#box1 div').html( $(window).scrollTop() + 'px vertikal,<br/>'  
            + $(window).scrollLeft() + 'px horizontal.' );  
    });  
});
```

Abmessungen und Positionen von Containern

Die Scroll-Position des Fensters lässt sich auf die gleiche Weise ermitteln:

Dazu werden in der Methode keine Parameter mit angegeben:

```
$(document).ready(function() {  
    $(window).scroll(function() {  
        $('#box1 div').html($(window).scrollTop() + 'px vertikal,<br/>' + $(window).scrollLeft() +  
            'px horizontal. ');  
    });  
});
```


Abmessungen und Positionen von Containern

Die Scroll-Position lässt sich auch innerhalb eines Elementes ermitteln bzw. lässt sich diese ansteuern:

Dazu müssen die Style-Definitionen angepasst werden.

```
#box2{  
    position:absolute; top:10px; left:50px;  
    ...  
    overflow:scroll; }
```

nun noch die jQuery-Codes:

```
$(document).ready(function() {  
    // Textabsätze in Box 2 schreiben:  
    for (var i=0;i<100;i++) {  
        $('<p>Textabsatz</p>').appendTo('#box2');  
    }  
    // Ausgabebox initialisieren:  
    $('#box1 div').html('<b>Scrollposition:</b><br/>' + $('#box2').scrollTop() + 'px vertikal');  
  
    // beim Scrollen von Box 2 neue Position in Box 1 schreiben:  
    $('#box2').scroll(function(){  
        $('#box1 div').html('<b>Position:</b><br/>' + $('#box2').scrollTop() + 'px vertikal');  
    });  
});
```

VIELEN DANK!

