

jQuery: Eigenes Plugin [Part 2: Erweiterte Parameter]

== Einführung == Dies ist die Fortsetzung zum ersten Teil <http://www.easy-coding.de/wiki/html-ajax-und-co/jquery-eigenes-plugin-part-1-grundlagen.html>, in dem die Grundlagen erklärt wurden. In diesem Teil erkläre ich, wie man erweiterte Parameter hinzufügt.

== Vorbereitung == Nehmen wir unsere Beispieldateien, vom vorherigen Teil.

plugin_first.js

Quellcode

```
1. $.fn.extend({  
2.   pluginOwn: function(pixel, zeit)  
3.   {  
4.     return this.each(function()  
5.     {  
6.       $(this).animate({top: '+=' + pixel + 'px'}, zeit);  
7.     });  
8.   }  
9. }  
10. }  
11. }  
12. });
```

Alles anzeigen

HTML:

Quellcode

```
1. <!DOCTYPE html>  
2. <html>  
3. <head>  
4. <title>Eigenes Plugin</title>  
5. <script language="javascript" type="text/javascript"  
6.   src="https://ajax.googleapis.com/ajax/libs/jquery/1.6.4/jquery.min.js"></script>  
7. <script language="javascript" type="text/javascript" src="plugin_first.js"></script>  
8. <style type="text/css">  
9. <!--  
10. #test_box {  
11.   width: 100px;  
12.   height: 100px;  
13.   background-color: red;  
14.   position: absolute;  
15. }  
16. -->  
17. </style>  
18. <script language="javascript" type="text/javascript">  
19. $(document).ready(function()  
20. {  
21.   $('#test_box').pluginOwn(250, 1250);  
22. }  
23. </script>  
24. </head>  
25. <body>  
26. <div id="test_box"></div>  
27. </body>  
28. </html>
```

Alles anzeigen

== Beginn == Nun fangen wir an. Zuerst wenden wir uns unserer JS-Datei zu. Um mehrere Parameter und vor allem default-Parameter setzen zu können übergeben wir keine einzelnen Variablen, sondern gleich ein ganzes Objekt, was wir

so machen, wenn unser Objekt 'paras' heißt:

Quellcode

```
1. $.fn.extend({  
2. pluginOwn: function(paras)  
4. {  
5. return this.each(function()  
6. {  
7. $(this).animate({top: '+='+pixel+'px'}, zeit);  
9. });  
10. }  
12.});
```

Alles anzeigen

== Default-Parameter == Momentan haben wir ein Objekt mit den Paramtern, die übergeben werden. Jetzt können wir auch noch Default-Paramter setzen, indem wir ein Objekt erstellen, das die Default-Parameter enthält und dieses anschließend mit unserem ersten Objekt zusammenführen:

Quellcode

```
1. $.fn.extend({  
2. pluginOwn: function(paras)  
4. {  
5. var defaults = {  
6. speed: 1500, //Diese Default-Werte werden  
7. art: top; //später zu unserem 'paras'-Objekt hinzugefügt,  
8. movement: '500'; //wenn 'paras' diese Eigenschaften noch nicht hat  
9. };  
10. var paras = $.extend(defaults, paras); //Zusammenführung der beiden Objekte  
12. return this.each(function()  
14. {  
15. $(this).animate({paras.art: '+='+paras.movement+}, paras.speed); //Normaler Zugriff auf die Eigenschaften eines  
Objekts  
16. });  
17. }  
18.});
```

Alles anzeigen

== Plugin nutzen == Da wir nun unser Plugin fertig haben, können wir es ausprobieren. Da es nun über Default-Werte verfügt gibt es verschiedenste Möglichkeiten:

Quellcode

```
1. <script language="javascript" type="text/javascript">  
2. $(document).ready(function()  
3. {  
4. //Nur die Eigenschaft 'speed' gesetzt. Da wir aber Defaultsettings haben sind jetzt 'art' top und 'movement' 500;  
5. $('#test_box').pluginOwn({speed: 1500});  
6. });  
7. </script>
```

Quellcode

```
1. <script language="javascript" type="text/javascript">  
2. $(document).ready(function()  
3. {  
4. $('#test_box').pluginOwn({movement: '250', art: left});
```

```
5. });
6. </script>
```

Quellcode

```
1. <script language="javascript" type="text/javascript">
2. $(document).ready(function()
3. {
4. $('#test_box').pluginOwn({art: top});
5. });
6. </script>
```