Jquery: Form validation+Rounded Corners+List Items - Developer News

NEWS_PDF_AUTHOR: kaotik

NEWS_PDF_DATE: 2009/8/26 16:00:00

This tutorial will teach you 4 things:

* How to apply rounded corners

* Form validation

* Passing variables into a list

* Customizing jquery plugins

First go to google code (http://code.google.com/p/jqueryxoops/downloads/list) and download
tutorial 12. This zip has all the files you need. They are from my previous tutorial.

Part 1 - Rounded corners

Let's start with rounded corners. Open test.php and replace all code with this:

```
#box { background-color:#FFFF99; border:thin solid #FF0000; width:70%; height:50px;}
#myformDiv { background-color:#FF9900; width:200px; height:70px;}
```

```
$(document).ready(function() { //Finish loading the entire page before processing any javascript

$('#myformDiv').corner();  // Rounds corners of selected div. In this example "myformDiv"

   $("#subBut").click(function(event) {
      var formContent = $("#form1").serialize();
      $("#box").load('myserv.php',formContent);
   });
});
```

Name

Ajax call

Here's the changes:
Line 2 - I created a new ID style called "myformDiv" gave it a background color, width and height
Line 8- Load our rounded corners plugin.
Line 13- This is where we set rounded corners. I've binded the jquery plugin "corners" to the selector myformDiv. You will notice the empty (). You can set the radius of the corner and even which corners you want to round. 1,2,3 or all 4 (this case) are possible.
Line 22 and 29 - I've created a div around my form. This is what gets rounded.

Easy wasn't it? let's move on.

Part 2 - Validation

Form validation is one of those things best done with javascript. Why come back to the server to check if a user has filled all requiered fields? Now, in order for form validation to also work with ajax, we have to combine both. Replace all code from test.php with this:

```
#box { background-color:#FFFF99; border:thin solid #FF0000; width:70%; height:50px;}
#myformDiv { background-color:#FF9900; width:300px; height:170px;}
.error-highlight {border: 2px solid #f00;}
.errMissFld {color: #f00;}
```

```
$(document).ready(function() { //Finish loading the entire page before processing any javascript
```

```
$('#myformDiv').corner();  // Rounds corners of selected div. In this example "myformDiv"

 $('#form1').bind('submit', function() {
   tst= $(this).validation();
   if(tst==false){return tst;}
      var formContent = $("#form1").serialize();
      $("#box").load('myserv.php',formContent);
      return false;
   });
});
```

Name

Name

Ajax call

Line 4 - Will create a red border around any form element that's requiered and isn't filled
Line 5 - make my error message red
Line 18- I had to change things a little from my previous example. In order for ajax and
validation to play nice with each other, I am now binding the submit button of the form "form1".
Line 19 - It will perform validation on all elements inside form "form1".
Line 20 - If validation failed, meaning a requiered field is empty, then it will stop executing
anything beyond this point and return. If, however validation was true then it will continue
processing the remaing code inside this function.
Line 30 and 35 - I've wrapped the fields I want to be requiered with

elements (could be div, etc) with a class of "required-field". Pleae note that this isn't the original jquery.validation plugin. It's been hacked by me to perform this way.

open myserv.php and replace all code with this:

Since I am now using 2 text fields, I've named them diferently, hence the change in myserv.php Moving on.

Part 3 - Passing variables

We are now going to create a list of text that allows users to click and then pass info to server through ajax. One of the advantages of using javascript over php is that we can take advantage of html code, what does this mean? Open test.php and replace all code with this:

```
#box { background-color:#FFFF99; border:thin solid #FF0000; width:70%; height:50px;}
#myformDiv { background-color:#FF9900; width:300px; height:170px;}
.error-highlight {border: 2px solid #f00;}
.errMissFld {color: #f00;}
```

```
$(document).ready(function() { //Finish loading the entire page before processing any javascript

$('#myformDiv').corner();  // Rounds corners of selected div. In this example "myformDiv"

 $('#form1').bind('submit', function() {
    tst= $(this).validation();
    if(tst==false){return tst;}
       var formContent = $("#form1").serialize();
       $("#box").load('myserv.php',formContent);
       return false;
   });

   $(".mylink").bind("click", function() {
```

```
    var hol=$(this).attr('myval');
    var formContent ="action=radio&link="+hol;
    $("#box").load('myserv.php',formContent);
    });

});
```

Name

Name

This is my prefered radio station

Ajax call

Now open myserv.php and replace all with this:

Now let's go through all the changes line by line starting with test.php
1 to 9 - I've created a php loop populate a list.
36 to 41 - I've created another click function. This will bind all links with class "mylink". You will notice something new here. I have a javascript variable called "hol" that is retreiving the content of an html attribute called "myval". You can create as many attributes as you like to pass information back to the server. On line 38 also notice that I'm creating formContent and adding more info into it. With php the concotenation symbol is "." but in javascript it is "+".
Line 67 - You can see a single example of how my link looks. Notice myval="12"
Line 70 to 74 - A php loop to populate more links. Since most of you understand php I won't explain this.
Now on to file myserv.php
Line 1- This a simple php IF statement You could also use this to run functions. So that you would have a php function for each ajax ajax.

I hope this has been helpfull. I've got a couple more in the works :)

This tutorial will teach you 4 things:

* How to apply rounded corners

* Form validation

* Passing variables into a list

* Customizing jquery plugins

First go to google code (http://code.google.com/p/jqueryxoops/downloads/list) and download tutorial 12. This zip has all the files you need. They are from my previous tutorial.

Part 1 - Rounded corners

Let's start with rounded corners. Open test.php and replace all code with this:

```
#box { background-color:#FFFF99; border:thin solid #FF0000; width:70%; height:50px;}
#myformDiv { background-color:#FF9900; width:200px; height:70px;}
```

```
$(document).ready(function() { //Finish loading the entire page before processing any javascript

$('#myformDiv').corner();  // Rounds corners of selected div. In this example "myformDiv"

    $("#subBut").click(function(event) {
        var formContent = $("#form1").serialize();
        $("#box").load('myserv.php',formContent);
    });
});
```

Name

Ajax call

Here's the changes:
Line 2 - I created a new ID style called "myformDiv" gave it a background color, width and height
Line 8- Load our rounded corners plugin.
Line 13- This is where we set rounded corners. I've binded the jquery plugin "corners" to the selector myformDiv. You will notice the empty (). You can set the radius of the corner and even which corners you want to round. 1,2,3 or all 4 (this case) are possible.
Line 22 and 29 - I've created a div around my form. This is what gets rounded.

Easy wasn't it? let's move on.

Part 2 - Validation

Form validation is one of those things best done with javascript. Why come back to the server to check if a user has filled all requiered fields? Now, in order for form validation to also work with ajax, we have to combine both. Replace all code from test.php with this:

```
#box { background-color:#FFFF99; border:thin solid #FF0000; width:70%; height:50px;}
#myformDiv { background-color:#FF9900; width:300px; height:170px;}
.error-highlight {border: 2px solid #f00;}
.errMissFld {color: #f00;}
```

```
$(document).ready(function() { //Finish loading the entire page before processing any javascript

$('#myformDiv').corner();  // Rounds corners of selected div. In this example "myformDiv"

 $('#form1').bind('submit', function() {
   tst= $(this).validation();
   if(tst==false){return tst;}
      var formContent = $("#form1").serialize();
      $("#box").load('myserv.php',formContent);
      return false;
```

```
    });
});
```

Name

Name

Ajax call

Line 4 - Will create a red border around any form element that's requiered and isn't filled
Line 5 - make my error message red
Line 18- I had to change things a little from my previous example. In order for ajax and validation to play nice with each other, I am now binding the submit button of the form "form1".
Line 19 - It will perform validation on all elements inside form "form1".
Line 20 - If validation failed, meaning a requiered field is empty, then it will stop executing anything beyond this point and return. If, however validation was true then it will continue processing the remaing code inside this function.
Line 30 and 35 - I've wrapped the fields I want to be requiered with

 elements (could be div, etc) with a class of "required-field". Pleae note that this isn't the original jquery.validation plugin. It's been hacked by me to perform this way.

open myserv.php and replace all code with this:

Since I am now using 2 text fields, I've named them diferently, hence the change in myserv.php
Moving on.

Part 3 - Passing variables

We are now going to create a list of text that allows users to click and then pass info to server through ajax. One of the advantages of using javascript over php is that we can take advantage of html code, what does this mean? Open test.php and replace all code with this:

```
#box { background-color:#FFFF99; border:thin solid #FF0000; width:70%; height:50px;}
#myformDiv { background-color:#FF9900; width:300px; height:170px;}
.error-highlight {border: 2px solid #f00;}
.errMissFld {color: #f00;}
```
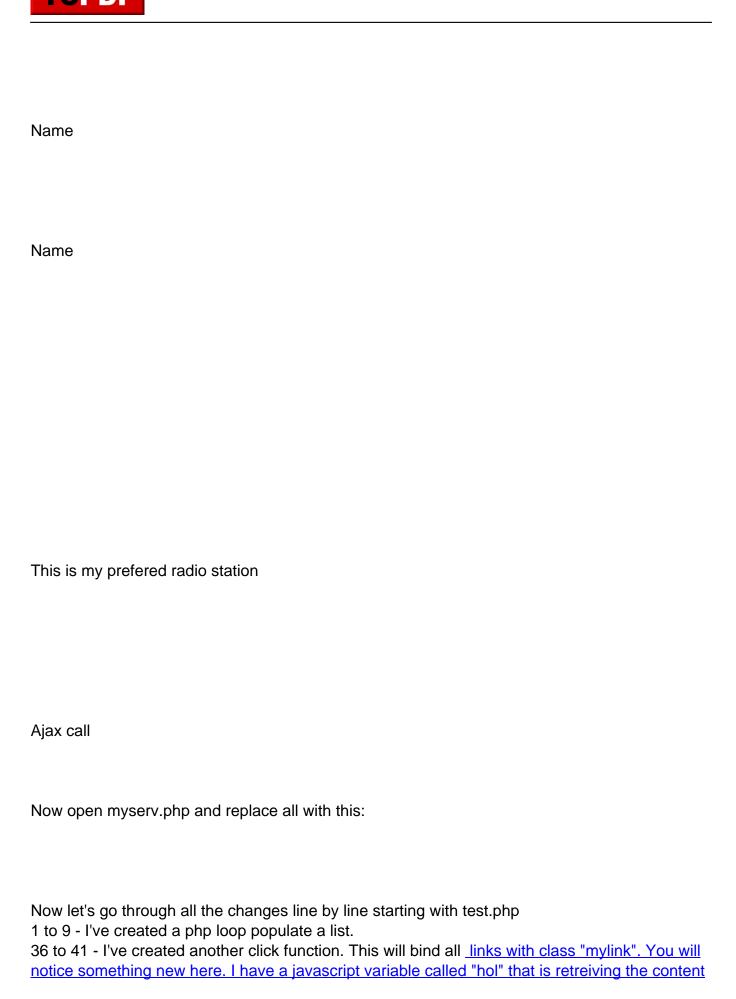
```
$(document).ready(function() { //Finish loading the entire page before processing any javascript

$('#myformDiv').corner();  // Rounds corners of selected div. In this example "myformDiv"

 $('#form1').bind('submit', function() {
   tst= $(this).validation();
   if(tst==false){return tst;}
      var formContent = $("#form1").serialize();
      $("#box").load('myserv.php',formContent);
      return false;
   });

   $(".mylink").bind("click", function() {
   var hol=$(this).attr('myval');
   var formContent ="action=radio&link="+hol;
   $("#box").load('myserv.php',formContent);
   });

});
```

Name

Name

This is my prefered radio station

Ajax call

Now open myserv.php and replace all with this:

Now let's go through all the changes line by line starting with test.php
1 to 9 - I've created a php loop populate a list.
36 to 41 - I've created another click function. This will bind all links with class "mylink". You will notice something new here. I have a javascript variable called "hol" that is retreiving the content

of an html attribute called "myval". You can create as many attributes as you like to pass information back to the server. On line 38 also notice that I'm creating formContent and adding more info into it. With php the concotenation symbol is "." but in javascript it is "+".
Line 67 - You can see a single example of how my link looks. Notice myval="12"
Line 70 to 74 - A php loop to populate more links. Since most of you understand php I won't explain this.
Now on to file myserv.php
Line 1- This a simple php IF statement You could also use this to run functions. So that you would have a php function for each ajax ajax.

I hope this has been helpfull. I've got a couple more in the works :)