Tutorial- Javascript+Jquery. A simple Beginning - Developer News

NEWS_PDF_AUTHOR: kaotik

NEWS_PDF_DATE: 2009/8/25 13:40:00

In this tutorial I will teach you the basics of javascript with jquery. This tutorial is especially geared towards php developers who have been hesitant to venture into javascript.
The question that pops up most often is: why use javascript if I already have PHP? Well, because PHP is a server side language while javascript resides on the client. Actions that are tacky done on PHP, such as table sorting are a breeze on javascript. Mind you that I'm not advocating you drop PHP, on the contrary! If you combine PHP and javascript you can take your code to a whole different level. Let's start.
Before we start, a couple of requirements for this tutorial:
- Go to jquery.com and grab jquery-1.3.2.min.js This is the minified version of jquery. It's great to start learning with. When you are proficient with javascript you might want to get the full version to perform your own hacks.
- Go to getfirebug.com and download firebug.js This is a great development tool that allows you to debug javascript directly through firefox (I'm planning a tutorial just dedicated to this).
-An editor either wysiwyg or ide to change php/html/etc files.
-A local php server. If you are not running one, go check xampp or similar.

First create an empty file called: test.php Inside place this code:

```
alert ("hello world");
```

This will popup an alert box that says hello world! Now since we understand php, let's bring the two together. Replace all previous code with this:

```
alert ("");
```

We have a php variable called $tst that get's passed into the alert box. Now let's bring in jquery and start making things more interesting. Replace all previous coe in file test.php with this:

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

```
 $(document).ready(function() {
//Finish loading the entire page before processing any javascript
 $("#subBut").click(function(event) {
 var formContent = $("#form1").serialize();
 $("#box").load('myserv.php',formContent);
 });
 });
```

```
 Name
```

```
 Ajax call
```

Now create another file called myserv.php and place this code inside:

Now test it! What we are doing here is creating an ajax response to our initial test.php file. It takes the form data and passes it to myserv.php through $_GET. The good thing about using jquery $.serialize (found on line 6) is that it will grab all elements in a form. So, even if you add more later on, it will still grab them.
Now a line by line explanation of the javscript code:
Lines 4 and 5 load jquery and firebug
Line 7 All code inside this function only gets processed AFTER the entire page has loaded. This is especially important for us php developers since we have php code many times scattered along our pages.
Line 8 is an event function. Meaning, when an event happens, run the code inside. In this case the event is a "click" and it has been binded to selector "subBut" which is the submit button in our form. So, when someone presses the submit button, the code inside gets executed.
Line 9 is a great jquery function called $.serialize It grabs all form elements inside our form "form1" and correctly formats it to be sent using $_GET to our response page. In order for this to work I created a javascript variable called "formContent". this holds the formated data. NOTE: All variables in javascript must first be declared as such before being used.
Line 10 This is where jquery shines. A simple line that does so much! The selector in this case is a div with id "box". This is where the ajax response will be loaded into.
load('myserv.php',formContent) will send the content of our javascript variable "formContent" to the page myserv.php which processes the information then sends back a response.

This is a simple tutorial to get you started. I have more on the way :)

In this tutorial I will teach you the basics of javascript with jquery. This tutorial is especially geared towards php developers who have been hesitant to venture into javascript.
The question that pops up most often is: why use javascript if I already have PHP? Well, because PHP is a server side language while javascript resides on the client. Actions that are tacky done on PHP, such as table sorting are a breeze on javascript. Mind you that I'm not advocating you drop PHP, on the contrary! If you combine PHP and javascript you can take your code to a whole different level. Let's start.
Before we start, a couple of requirements for this tutorial:
- Go to jquery.com and grab jquery-1.3.2.min.js This is the minified version of jquery. It's great to start learning with. When you are proficient with javascript you might want to get the full version to perform your own hacks.
- Go to getfirebug.com and download firebug.js This is a great development tool that allows you to debug javascript directly through firefox (I'm planning a tutorial just dedicated to this).
-An editor either wysiwyg or ide to change php/html/etc files.
-A local php server. If you are not running one, go check xampp or similar.

First create an empty file called: test.php Inside place this code:

```
alert ("hello world");
```

This will popup an alert box that says hello world! Now since we understand php, let's bring the two together. Replace all previous code with this:

```
alert ("");
```

We have a php variable called $tst that get's passed into the alert box. Now let's bring in jquery and start making things more interesting. Replace all previous coe in file test.php with this:

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

```
  $(document).ready(function() {
//Finish loading the entire page before processing any javascript
  $("#subBut").click(function(event) {
  var formContent = $("#form1").serialize();
  $("#box").load('myserv.php',formContent);
  });
  });
```

Name

Ajax call

Now create another file called myserv.php and place this code inside:

Now test it! What we are doing here is creating an ajax response to our initial test.php file. It takes the form data and passes it to myserv.php through $_GET. The good thing about using jquery $.serialize (found on line 6) is that it will grab all elements in a form. So, even if you add more later on, it will still grab them.
Now a line by line explanation of the javscript code:
Lines 4 and 5 load jquery and firebug
Line 7 All code inside this function only gets processed AFTER the entire page has loaded. This is especially important for us php developers since we have php code many times scattered along our pages.
Line 8 is an event function. Meaning, when an event happens, run the code inside. In this case the event is a "click" and it has been binded to selector "subBut" which is the submit button in our form. So, when someone presses the submit button, the code inside gets executed.
Line 9 is a great jquery function called $.serialize It grabs all form elements inside our form "form1" and correctly formats it to be sent using $_GET to our response page. In order for this to work I created a javascript variable called "formContent". this holds the formated data. NOTE: All variables in javascript must first be declared as such before being used.
Line 10 This is where jquery shines. A simple line that does so much! The selector in this case is a div with id "box". This is where the ajax response will be loaded into.
load('myserv.php',formContent) will send the content of our javascript variable "formContent" to the page myserv.php which processes the information then sends back a response.

This is a simple tutorial to get you started. I have more on the way :)