

amplio-Vita quickEdit Tutorial

quickEdit is a library that enables you to edit any kind of text in your website very easily. Let's see a working example and understand how easily quickEdit can be implemented. In this example I'll show you how to apply quickEdit to `` elements but you can use these with any text based element or any image element you want. The method is completely the same. Download [this](#) and unzip it. Look at the quickEditInitial.htm file.

In this very simple example we have a style.css file and some HTML codes. We will add our quickEdit library and make the element with `id="itemCount"` editable.

We implement our quickEdit library by adding these five js files. The functions in these files are categorized. You can also download the file aV.all.js if you do not want to add these files separately.

```
1: <script src="aV.main.ajax.js" type="text/javascript"></script>
2: <script src="aV.ext.string.js" type="text/javascript"></script>
3: <script src="aV.main.events.js" type="text/javascript"></script>
4: <script src="aV.main.visual.js" type="text/javascript"></script>
5: <script src="aV.plg.quickEdit.js" type="text/javascript"></script>
```

Note: In our Live example we added `src="../../js/aV.main.ajax.js"`. If you also want to organize your libraries in a folder you should change this source accordingly. This example assumes that all of the files are in the same directory.

We add these entries to the head section since we will use the function **QuickEdit.init** as soon as the page is finished loading. We can accomplish this by editing our body tag from:

```
1: <body>
```

to:

```
1: <body onload="QuickEdit.init()">
```

Now actually, everything is ready. We just add the parameters that will make our element an editable element. Find the element with the `id="itemCount"` which is

```
1: <span id="itemCount">2</span>
```

We will change it as follows :

```
1: <span id="itemCount" editcond="1" editaction="itemCountChange.php"
  editparams="itemId=1&itemCount=">2</span>
```

Let's examine what those parameters mean and how they work.

editcond

The first parameter `editcond` is used when the editability of the element depends on some variables. In this case I wanted it to be always editable but it could have been as follows `editcond="isAdmin"` where `isAdmin` is a globally defined javascript variable name. This may be used to make the element editable only for users who has administrative rights.

Note: Do not forget that this condition is checked only client-side, you should always check user rights or in short editability rights from your server-side script.

editaction

The second parameter `editaction` is the server-side script files address that will be called when the editing is performed just as a form's action property. There will be an AJAX request working behind to *actually* change the value. This lets you to control and apply the change on the server side. The server-side script file may be used to store the new value in a database or XML file on the server. The parameters to this file can be given by the third parameter.

editparams

The third parameter, `editparams` keeps the parameters that will be given to the server-side script file which is defined in `editaction` property. In this example I gave an `itemId=1` parameter and `itemCount=`. So in this case two parameters will be sent to the php file. Writing only the parameter name at the end of the parameters means that the new value of the edited text will be assigned to that parameter name. In this case the parameter `itemCount` will hold the value which the user entered instead of 2.

QuickEdit has two more parameters that I believe you will like :

edittype : Adding the parameter *edittype="TEXTAREA"* will make a multiline edit which means that your box will expand in y direction. Or in technical terms, there will be a TEXTAREA element instead of a INPUT TYPE="text" element when user clicks to change the text.

editfade : Giving this parameter will make the element fade to the given value when it is on hover. Giving 0.5 to this parameter will make the element 50% transparent when the mouse is over the element. When it is evaluated as "editable" from the **editcond** property of course.

Also add these parameters and see what happens. May be you'll like it ;)

A note about styles:

When the editable element is hovered its class will change to `editableLabel`. By writing a css for this class you can change the view of the editable object as much as you like. Add this to the end of the style.css file and see the change if you like :

```
1: .editableLabel
2: {
3:   border: #000000 1px dashed;
4: }
5: .editableLabel input
6: {
7:   border: none;
8:   padding: 0px;
9:   margin: 0px;
10:  display: inline;
11:  width: 30px!important;
12: }
```

Final version can be seen from <http://amplio-vita.net/JSlib/demos/quickEdit/quickEditFinal.htm>

If you have any questions please mail to :

byk@amplio-vita.net
snlzkn@amplio-vita.net

Please do not hesitate to ask or request anything. Only this can help us to improve Life!