


QuickTags

Table of contents

- [build a library of reusable tags](#)
 - [RFEs](#)
 - [examples](#)
 - [categories](#)
 - [Forms](#)
 - [text area](#)
 - [rolldown](#)
 - [option library](#)
 - [library of scripts](#)
 - [checkboxes](#)
 - [radio buttons](#)
-

build a library of reusable tags

The quicktags are very cool and now support categories by object... though I'm not sure what objects specifically as the objects supported are not loaded from the db... they are entered by hand 

RFEs

- the categories for the quick tags is vague... needs a list of cats or a drop down to help users understand options here
 - note : categories only work one at a time. This won't work



- support for html tags
- support for export to share among other libraries

examples

The quicktags are those little icons on the right, since 1.8rc2, of the edit page : you click on it and a javascript is used to insert a tag in the textarea, depending your actual selection. Given a good browser like [mozilla](#) you can select text then click on the quick tag and your text will be wrapped in the desired tag! This has enormous implications for grading papers on line!

early message from mose:

In 1.8rc2 I introduce `tiki-admin_quicktags.php` (link in admin menu, named QuickTags, accessible by admin only) you can customise the tags names, inserts and icons to fit your own needs. It can be used for redundant formatting insertion, quoting, using in a virtual classroom to annotate contents by dgd, anything.

QuickTags are now stored in the very simple table `tiki_quicktags`. Can be enhanced.

categories

You must add a category for each of your quickags. But only one category per quicktag is now supported: This won't work.



Here are some quicktags I would use to create forms:

Forms

here's the short answer fill in the blank



```
<form name="test_form"><input name="testfield" value="Wiki form Builder"></form>
```

text area

Here's a working text area... the tags required for creating this text area field could be created using QuickTags (unfortunately, right now, there's a bug in the QuickTags that doesn't allow more than 21 quick tags... other wise I'd create demos of all of these tags...



```
<textarea cols="80" rows="5" name="text area">
```

Instructions for the assignment can go in here or templates for the students to use while writing?</textarea>

rolldown

need those rolldowns:

I lifted this straight from the W3C site on HTML forms... mozilla's composer under win2k is buggy, preview doesn't work.



```
<select name="Verb">
<option selected="selected" label="none" value="none">Click the rolldown</option>
<optgroup label="Verbs">
<option label="3.7.1" value="pm3_3.7.1">Past Tense</option>
<option label="3.7" value="pm3_3.7">Present</option>
```

```
<option label="3.5" value="pm3_3.5">Future</option>
</optgroup><optgroup label="Adjectives">
<option label="3.7" value="pm2_3.7">green</option>
<option label="3.5" value="pm2_3.5">yellow</option>
</optgroup><optgroup label="Averbs">
<option label="3.7R" value="IRX_3.7R">quickly</option>
<option label="3.5R" value="IRX_3.5R">slowly</option>
</optgroup>
</select>
</p>
```

option library

Imagine having to retype options like this one. This is why the QuickTags can be very useful and why the use of wiki form templates are critical for speeding development and delivery of content and assignments. Ideally there would be a way of knowing what templates are being used at any one time but that's a future problem.

```
<select name=grade >
<option selected="selected" label="none" value="none">Assign point value for this test...</option>
<option value="100" selected>100</option>
<option value="99">99</option>
<option value="98">98</option>
<option value="97">97</option>
<option value="96">96</option>
<option value="95">95</option>
<option value="94">94</option>

<option value="93">93</option>
<option value="92">92</option>
<option value="91">91</option>
<option value="90">90</option>
<option value="89">89</option>
<option value="88">88</option>

<option value="87">87</option>
<option value="86">86</option>
<option value="85">85</option>
<option value="84">84</option>
<option value="83">83</option>
<option value="82">82</option>

<option value="81">81</option>
<option value="80">80</option>
<option value="79">79</option>
<option value="78">78</option>
<option value="77">77</option>
<option value="76">76</option>

<option value="75">75</option>
```

<option value="74">74</option>
<option value="73">73</option>
<option value="72">72</option>
<option value="71">71</option>
<option value="70">70</option>

<option value="69">69</option>
<option value="68">68</option>
<option value="67">67</option>
<option value="66">66</option>
<option value="65">65</option>
<option value="64">64</option>

<option value="63">63</option>
<option value="62">62</option>
<option value="61">61</option>
<option value="60">60</option>
<option value="59">59</option>
<option value="58">58</option>

<option value="57">57</option>
<option value="56">56</option>
<option value="55">55</option>
<option value="54">54</option>
<option value="53">53</option>
<option value="52">52</option>

<option value="51">51</option>
<option value="50">50</option>
<option value="49">49</option>
<option value="48">48</option>
<option value="47">47</option>
<option value="46">46</option>

<option value="45">45</option>
<option value="44">44</option>
<option value="43">43</option>
<option value="42">42</option>
<option value="41">41</option>
<option value="40">40</option>

<option value="39">39</option>
<option value="38">38</option>
<option value="37">37</option>
<option value="36">36</option>
<option value="35">35</option>
<option value="34">34</option>

<option value="33">33</option>

```
<option value="32">32</option>
<option value="31">31</option>
<option value="30">30</option>
<option value="29">29</option>
<option value="28">28</option>

<option value="27">27</option>
<option value="26">26</option>
<option value="25">25</option>
<option value="24">24</option>
<option value="23">23</option>
<option value="22">22</option>

<option value="21">21</option>
<option value="20">20</option>
<option value="19">19</option>
<option value="18">18</option>
<option value="17">17</option>
<option value="16">16</option>

<option value="15">15</option>
<option value="14">14</option>
<option value="13">13</option>
<option value="12">12</option>
<option value="11">11</option>
<option value="10">10</option>

<option value="9">9</option>
<option value="8">8</option>
<option value="7">7</option>
<option value="6">6</option>
<option value="5">5</option>
<option value="4">4</option>

<option value="3">3</option>
<option value="2">2</option>
<option value="1">1</option>
<option value="0">0</option>
</select>
```

library of scripts

these fields could be supplied from a simple text or wiki file like this:



```
i = 'cat yourrubricfile.txt';
for n in i;
```

```
do grep $n form.txt;
done
```

Or maybe not... just a thought
this doesn't work



```
{picture file=img/wiki_up/quicktags.png width=300 height=300 align=center}
```

this does



checkboxes

Check boxes are good in forms

```
<input type="checkbox"      <input type="checkbox" or here<input      or there<input
name="this checkbox"    name="this checkbox" type="checkbox"      type="checkbox"
value="sessions">click here value="sessions"> name="this checkbox" name="this checkbox"
value="sessions">      value="sessions">
```

radio buttons

And how could we tune our pages to the right tone without radio buttons?

```
<input type="radio"    <input type="radio"    <input type="radio"    <input type="radio"
name="grade0"          name="grade0"          name="grade0"          name="grade0"
value="1">             value="1">             value="1">             value="1">
```