

## Introduction

- Macro
- Forms
- Aha Property (Drupal 6) – Ajax Framework (Drupal 7)

## Introduction to Jquery

- What is Jquery?
  - Javascript Library
  - Reduce Javascript code
  - Crossed browser
  - Visual affects
- The only function in JQuery- \$
  - \$('a.classname') **Selector**
  - \$('<div>new content</div>') **DOM Creation**
  - \$.browser.msie **Browser Detection**
    - Not supported in 7
- Jquery as a CSS Selector – Reason why it is popular
  - \$('.title') by class
  - \$('#main') by id
  - \$('.sidebar a, .content a'); child elements; multiple element
  - \$('input[type=text]'); css3 select attribute
  - \$('img[src\$=.png]') finding images
- Difference between versions
  - \$('input[type=text]'); Drupal 6/7 (jQuery 1.2.6/1.4x)
  - \$('input[type=text]'); Drupal 5 (jQuery 1.0.4)
- JQuery Effects
  - \$('h1').hide();
  - \$('h1').hide('slow');
  - \$('h1').hide(3000); 3 second to complete , jquery uses millisecond
  - \$('h1').fadeOut('slow');
  - \$('h1').slideUp('slow');
  - Toggle Events
    - \$('h1').slideToggle();
    - \$('h1').toggleClass('class-name');
  - Class Manipulation
    - \$('h1').addClass('class-name');
    - \$('h1').removeClass('class-name');
- JQuery Chainability
  - Multiple Operations on the same element
    - \$('h1').prepend('Hello').append('!').fadeOut().fadeIn();
      - All operations will occur on h1 tag
- Jquery Events
  - Trigger an action in response to the user
  - Most common events:
    - Adding Event
      - .click(functionName); when user clicks

Created with



download the free trial online at [nitropdf.com/professional](http://nitropdf.com/professional)

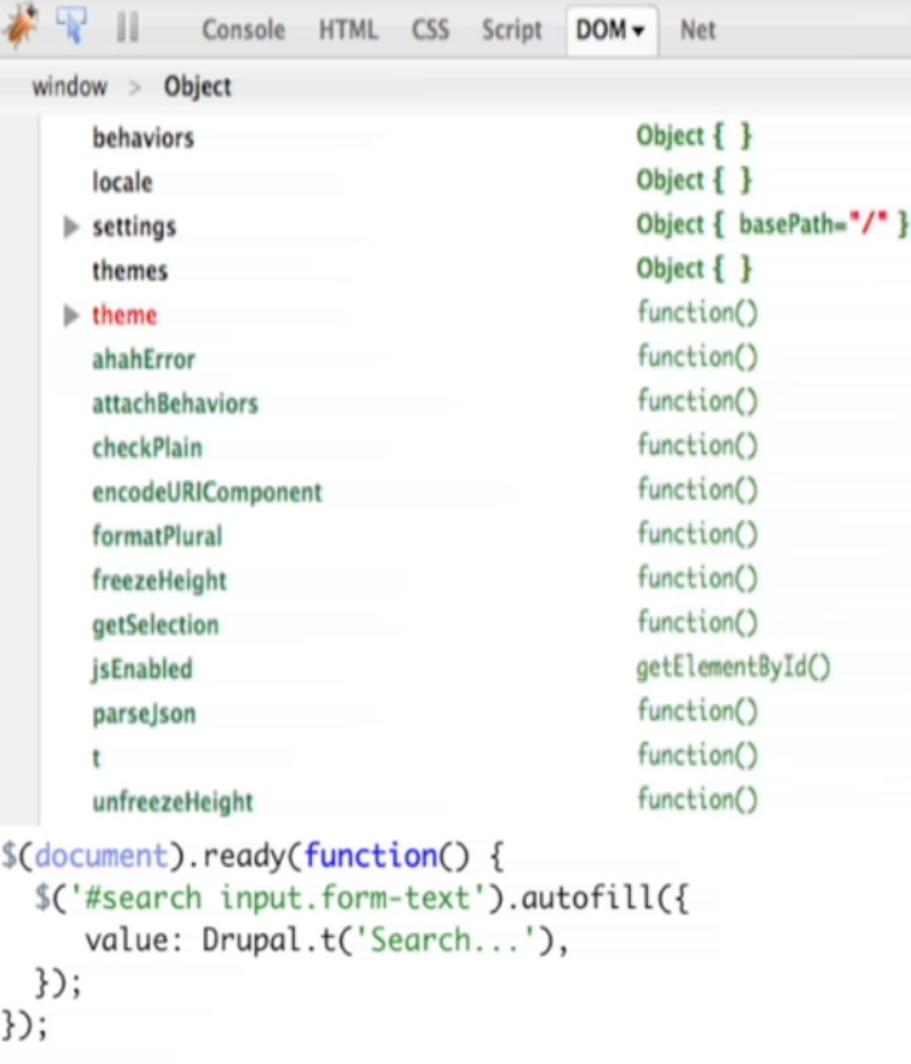
- `.change(functionName);` for radio buttons or checkboxes
- 
- `.toggle(function1, function2);`
- `.hover(function1, function2);` rolling the mouse over
  - Function name is the function to be executed
- Removing Event
  - `$(‘a’).unbind(‘click’, functionName)`

## Getting Hands On

- Exercise 1
  - Download a ready made Library
    - Place it in your theme directory
      - Create a js folder optional
  - Add scripts in .info
    - Scripts[] = js/library.js
  - Clear Cached data
  - To check if it is loaded
    - Right click and view source
    - As soon as you include a library, Drupal will add two Javascript and jquery
      - Jquery.js and Drupal.js
  - Read the library documentation on how to use particular plugin
  - Create another js file inside your js folder
    - Paste in the js selector and effects of that particular plugin
      - Replace the selector in your function to match your css
      - To test the code: use firebug console
        - Paste it in the right pane and see if it works

```
$(document).ready(function() {
  $('#search input.form-text').autofill({
    value: 'Search...'
  });
});
```

- Name it and add it into .info file
- Drupal .t (multi language functionality)
  - Collection of special Javascript function



The screenshot shows the DOM tab of a browser developer tools interface. The left sidebar lists properties of the window object, while the main pane displays a script block.

```
$(document).ready(function() {
  $('#search input.form-text').autocomplete({
    value: Drupal.t('Search...'),
  });
});
```

## Chapter 4 The Macro Maker Module

Created with



## Macros

[List](#)[Add macro](#)

### Title

⊕ Company Phone

⊕ Company Address

⊕ Today's Date

⊕ Kent Bye

⊕ Nate Haug

⊕ Jeff Robbins

[Add another](#)

### Title: \*

Abbas Hos Patria Volutpat Euismod Si 877-LULLABOT

► [Menu settings](#)

### Body:

node (story) – Euismod tum quae hendrerit. Lucidus pecus abdo abigo ad venio.

Caecus persto bene blandit neo. Iusto sagaciter ideo in torqueo validus premo praesent refero tu.

Facilisis minim ad eum dolor populus caecus. Iriure abbas olim. Melior nibh aliquip imputo inhibe. Magna eum importunus vel volutpat. Commoveo adipiscing vicis. Dignissim jugis ex ullamcorper. Consectetuer singularis os pala ille abdo nimis pneum. Abico ymo obruo occuro appellatio ibidem esca.

Pecus jus voco defui nunc fere letalis. Jumentum jugis dolor quadrum cogo facilisi. Exputo anteha. Velit autem abdo vereor. Tamen esse distineo et hos ullamcorper qui. Praesent incassum hos imperto tation eu usitas os eum. Lenis caecus dignissim et humo jugis luctus usitas illum pagus.

[Split summary at cursor](#)



Created with

 nitro<sup>PDF</sup><sup>®</sup> professional

download the free trial online at [nitropdf.com/professional](http://nitropdf.com/professional)



```
1 <?php
2 // $Ids
3
4 /**
5 * @file
6 * A jQuery-enhanced interface for inserting snippets.
7 */
8
9 /**
10 * Implementation of hook_menu().
11 */
12 function macromaker_menu() {
13   $items = array();
14
15   $items['admin/settings/macromaker'] = array(
16     'title' => 'Macros',
17     'description' => 'Create text macros that can be used on textareas and text fields.',
18     'page callback' => 'drupal_get_form',
19     'page arguments' => array('macromaker_settings_form'),
20     'access arguments' => array('administer site configuration'),
21   );
22
23   $items['admin/settings/macromaker/list'] = array(
24     'title' => 'List',
25     'description' => 'Create text macros that can be used on textareas and text fields.',
26     'page callback' => 'drupal_get_form',
27     'page arguments' => array('macromaker_settings_form'),
28     'access arguments' => array('administer site configuration'),
29     'weight' => 0,
30     'type' => MENU_DEFAULT_LOCAL_TASK,
31   );
32
33   $items['admin/settings/macromaker/add'] = array(
34     'title' => 'Add macro',
35     'description' => 'Add another macro.',
36     'page callback' => 'drupal_get_form',
37     'page arguments' => array('macromaker_edit_form'),
38     'access arguments' => array('administer site configuration'),
39     'weight' => 5,
40     'type' => MENU_LOCAL_TASK,
41   );
42 }
```

```
40     $items['admin/settings/macromaker/edit'] = array(
41       'title' => 'Edit macro',
42       'page callback' => 'drupal_get_form',
43       'page arguments' => array('macromaker_edit_form'),
44       'access arguments' => array('administer site configuration'),
45       'weight' => 5,
46       'type' => MENU_CALLBACK,
47   );
48
49   return $items;
50 }
```

```
52 /**
53  * Implementation of hook_theme().
54 */
55 function macromaker_theme() {
56   $items = array();
57
58   $items['macromaker_settings_form'] = array(
59     'arguments' => array('form' => NULL),
60   );
61
62   $items['macromaker_popup'] = array(
63     'arguments' => array('items' => array()),
64   );
65
66   return $items;
67 }
```

```
69 /**
70  * Implementation of hook_perm()
71 */
72 function macromaker_perm() {
73   return array('use macros');
74 }
```

```

77 * Menu callback for admin/settings/macromaker. Show a list of all
78 */
79 function macromaker_settings_form() {
80   $macros = variable_get('macromaker_macros', array());
81   $form = array();
82
83   $form['macros']['#tree'] = TRUE;
84   foreach ($macros as $mid => $macro) {
85     $form['macros'][$mid]['delete'] = array(
86       '#type' => 'checkbox',
87     );
88     $form['macros'][$mid]['title'] = array(
89       '#value' => $macro['title']
90     );
91     $form['macros'][$mid]['mid'] = array(
92       '#type' => 'hidden',
93       '#value' => $mid,
94     );
95     // This field will be hidden by javascript.
96     $form['macros'][$mid]['position'] = array(
97       '#type' => 'textfield',
98       '#default_value' => $mid,
99       '#size' => 3,
100      '#maxlength' => 4,
101      '#attributes' => array('class' => 'macro-position'),
102    );
103
104    $form['macros'][$mid]['edit'] = array(
105      '#value' => l(t('edit'), 'admin/settings/macromaker/edit/' . $mid),
106    );
107
108   $form['submit'] = array(
109     '#type' => 'submit',
110     '#value' => t('Submit'),
111   );
112
113   return $form;
114 }

```

```

117 /**
118 * Theme the display of the settings form as a table.
119 */
120 function theme_macromaker_settings_form($form) {
121   $headers = array(t('Title'), t('Position'), t('Edit'), t('Delete'));
122   $rows = array();
123   foreach (element_children($form['macros']) as $mid) {
124     $row = array();
125     $row[] = drupal_render($form['macros'][$mid]['title']);
126     $row[] = drupal_render($form['macros'][$mid]['position']);
127     $row[] = drupal_render($form['macros'][$mid]['edit']);
128     $row[] = drupal_render($form['macros'][$mid]['delete']);
129     $rows[] = $row;
130   }
131
132   if (empty($rows)) {
133     $rows[] = array(array('colspan' => 4, 'data' => t('There are currently no macros.')));
134   }
135
136   $rows[] = array(array('colspan' => 4, 'data' => l(t('Add another'),
137     'admin/settings/macromaker/add')));
138
139   $output = '';
140   $output .= theme('table', $headers, $rows, array('id' => 'macromaker-macros'));
141   $output .= drupal_render($form);

```



```

142   return $output;
143 }
144
145 /**
146 * Submit handler for the macromaker_settings_form().
147 */
148 function macromaker_settings_form_submit($form, $form_state) {
149   usort($form_state['values']['macros'], 'macromaker_sort');
150
151   $macros = variable_get('macromaker_macros', array());
152   $ordered = array();
153   $deleted = FALSE;
154   foreach ($form_state['values']['macros'] as $macro) {
155     $mid = $macro['mid'];
156     if (!$macro['delete']) {
157       $ordered[$mid] = $macros[$mid];
158     }
159     else {
160       unset($macros[$mid]);
161       $deleted = TRUE;
162     }
163   }

```

```

164 // In the event that a macro has been added while the form was being
165 $ordered += $macros;
166
167 variable_set('macromaker_macros', $ordered);
168
169 if ($ordered != $macros) {
170   drupal_set_message(t('The macro order has been saved.'));
171 }
172 if ($deleted) {
173   drupal_set_message(t('The checked macros have been deleted.'));
174 }
175 if (!$deleted && $ordered == $macros) {
176   drupal_set_message(t('No changes have been made.'));
177 }
178 }

181 * Form to add or edit a macro.
182 */
183 function macromaker_edit_form($form_state, $mid = NULL) {
184 if ($mid != NULL) {
185   $macros = variable_get('macromaker_macros', array());
186   $macro = isset($macros[$mid]) ? $macros[$mid] : NULL;
187 }
188
189 if (!isset($macro)) {
190   // on the "add" form, we use an empty array
191   $macro = array('title' => '', 'content' => '');
192   $mid = NULL;
193 }
194
195 $form = array();
196 $form['mid'] = array(
197   '#type' => 'value',
198   '#value' => $mid,
199 );
200 $form['title'] = array(
201   '#type' => 'textfield',
202   '#required' => 1,
203   '#title' => t('Macro title'),
204   '#description' => t('This is the text that a user will click to insert the text below
allowed.'),
205   '#default_value' => $macro['title'],
206 );

```



```

207 $form['content'] = array(
208   '#type' => 'textarea',
209   '#required' => 1,
210   '#title' => t('Macro content'),
211   '#description' => t('This is the text that will be inserted when the user cli
212   '#default_value' => $macro['content'],
213 );
214 if (!module_exists('token')) {
215   $form['content']['#description'] .= ' ' . t(' Optionally, you may install <a
216 href="http://drupal.org/project/token">Token module</a> to insert placeholders
217 }
218 else {
219   $form['tokens'] = array(
220     '#type' => 'fieldset',
221     '#title' => t('Tokens'),
222     '#description' => theme('token_help', array('global', 'user')),
223     '#collapsible' => TRUE,
224     '#collapsed' => TRUE,
225   );
226 }
227 $form['submit'] = array(
228   '#type' => 'submit',
229   '#value' => t('Submit'),
230 );
231 return $form;
232 }

```

```

234 /**
235 * Submit function for macromaker_edit_form()
236 */
237 function macromaker_edit_form_submit($form, &$amp;form_state) {
238   $macros = variable_get('macromaker_macros', array());
239   $title = $form_state['values']['title'];
240   $content = $form_state['values']['content'];
241   $macro = array('title' => $title, 'content' => $content);
242   $mid = $form_state['values']['mid'];
243   if ($mid) {
244     $macros[$mid] = $macro;
245     drupal_set_message(t('Macro has been updated.'));
246   }
247   else {
248     $macros[] = $macro;
249     drupal_set_message(t('New macro has been saved.'));
250   }
251   variable_set('macromaker_macros', $macros);
252   $form_state['redirect'] = 'admin/settings/macromaker';
253 }

```

```

255 /**
256 * Theme function to generate popup
257 */
258 function theme_macromaker_popup($items) {
259   $output = '<div class="macro-popup">';
260   $output .= '<div class="macro-title">' . t('macros') . '</div>';
261   $list = '';
262   foreach($items as $mid => $item) {
263     // the "rel" attribute acts as a lookup for the complete string
264     $list .= '<li><a href="#" rel="' . $mid . '">' . $item['title'] . '</a></li>';
265   }
266   if (strlen($list)) {
267     $output .= "<ul> $list </ul>";
268   }
269   $output .= '</div>';
270   return $output;
271 }

```

```

172 if ($deleted) {
173   drupal_set_message(t('The checked macros have been deleted.'));
174 }
175 if (!$deleted && $ordered === $macros) {
176   drupal_set_message(t('No changes have been made.'));
177 }
178 }

179 /**
180 * Form to add or edit a macro.
181 */
182
183 function macromaker_edit_form($form_state, $mid = NULL) {
184   if ($mid != NULL) {
185     $macros = variable_get('macromaker_macros', array());
186     $macro = isset($macros[$mid]) ? $macros[$mid] : NULL;
187   }

188   if (!isset($macro)) {
189     // on the "add" form, we use an empty array
190     $macro = array('title' => '', 'content' => '');
191     $mid = NULL;
192   }

```

```

195   $form = array();
196   $form['mid'] = array(
197     '#type' => 'value',
198     '#value' => $mid,
199   );
200   $form['title'] = array(
201     '#type' => 'textfield',
202     '#required' => 1,
203     '#title' => t('Macro title'),
204     '#description' => t('This is the text that a user will click to insert the text below. T
205 allowed.'),
206     '#default_value' => $macro['title'],
207   );
208   $form['content'] = array(
209     '#type' => 'textarea',
210     '#required' => 1,
211     '#title' => t('Macro content'),
212     '#description' => t('This is the text that will be inserted when the user clicks the tit
213     '#default_value' => $macro['content'],
214   );
215   if (!module_exists('token')) {
216     $form['content']['#description'] .= ' ' . t(' Optionally, you may install <a
217 href="http://drupal.org/project/token">Token module</a> to insert placeholders into macros
218   ');

```



```
217     }
218     $form['tokens'] = array(
219       '#type' => 'fieldset',
220       '#title' => t('Tokens'),
221       '#description' => theme('token_help', array('global', 'user')),
222       '#collapsible' => TRUE,
223       '#collapsed' => TRUE,
224     );
225   }
226
227   $form['submit'] = array(
228     '#type' => 'submit',
229     '#value' => t('Submit'),
230   );
231   return $form;
232 }
```

```
234 /**
235 * Submit function for macromaker_edit_form()
236 */
237 function macromaker_edit_form_submit($form, &$form_state) {
238   $macros = variable_get('macromaker_macros', array());
239   $title = $form_state['values']['title'];
240   $content = $form_state['values']['content'];
241   $macro = array('title' => $title, 'content' => $content);
242   $mid = $form_state['values']['mid'];
243   if ($mid) {
244     $macros[$mid] = $macro;
245     drupal_set_message(t('Macro has been updated.'));
246   }
247   else {
248     $macros[] = $macro;
249     drupal_set_message(t('New macro has been saved.'));
250   }
251   variable_set('macromaker_macros', $macros);
252   $form_state['redirect'] = 'admin/settings/macromaker';
253 }
```

```

255 /**
256 * Theme function to generate popup
257 */
258 function theme_macromaker_popup($items) {
259   $output = '<div class="macro-popup">';
260   $output .= '<div class="macro-title">' . t('macros') . '</div>';
261   $list = '';
262   foreach($items as $mid => $item) {
263     // the "rel" attribute acts as a lookup for the complete string
264     $list .= '<li><a href="#" rel="' . $mid . '">' . $item['title'] . '</a></li>';
265   }
266   if (strlen($list)) {
267     $output .= "<ul>" . $list . "</ul>";
268   }
269   $output .= '</div>';
270   return $output;
271 }

```

## Chapter 5 Adding Javascript to Module

- Drupal\_add\_js
  - Generalize Drupal Javascript function

### drupal\_add\_js

```

5 common.inc drupal_add_js($data = NULL, $type = 'module', $scope =
                           'header', $defer = FALSE, $cache = TRUE)
6 common.inc drupal_add_js($data = NULL, $type = 'module', $scope =
                           'header', $defer = FALSE, $cache = TRUE, $preprocess =
                           TRUE)
7 common.inc drupal_add_js($data = NULL, $options = NULL)
8 common.inc drupal_add_js($data = NULL, $options = NULL)

```

- - Search Drupal\_add\_js at api.drupal.org

## Parameters

**\$data:** (optional) If given, the value depends on the \$type parameter:

- 'core', 'module' or 'theme': Path to the file relative to `base_path()`.
- 'inline': The JavaScript code that should be placed in the given scope.
- 'setting': An array with configuration options as associative array. The array is directly placed in `Drupal.settings`. You might want to wrap your actual configuration settings in another variable to prevent the pollution of the `Drupal.settings` namespace.

**\$type:** (optional) The type of JavaScript that should be added to the page. Allowed values are 'core', 'module', 'theme', 'inline' and 'setting'. You can, however, specify any value. It is treated as a reference to a JavaScript file. Defaults to 'module'.

**\$scope:** (optional) The location in which you want to place the script. Possible values are 'header' and 'footer' by default. If your theme implements different locations, however, you can also use these.

**\$defer:** (optional) If set to TRUE, the defer attribute is set on the `<script>` tag. Defaults to FALSE. This parameter is not used with `$type == 'setting'`.

**\$cache:** (optional) If set to FALSE, the JavaScript file is loaded anew on every page call, that means, it is not cached. Defaults to TRUE. Used only when `$type` references a JavaScript file.

- **\$noprocess:** (optional) Should this JS file be annotated if this feature has been turned
  - [http://api.drupal.org/api/drupal/includes%21common.inc/function/drupal\\_add\\_js/6](http://api.drupal.org/api/drupal/includes%21common.inc/function/drupal_add_js/6)

```
* Implementation of hook_theme().
*/
function macromaker_theme() {
  $items = array();

  $items['macromaker_settings_form'] = array(
    'arguments' => array('form' => NULL),
  );

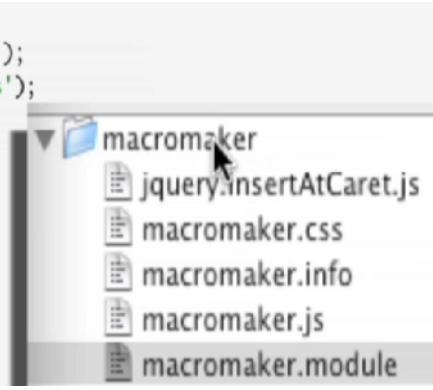
  $items['macromaker_popup'] = array(
    'arguments' => array('items' => array()),
  );

  return $items;
}
```

```
function macromaker_init() {
  $path = drupal_get_path('module', 'macromaker');
  drupal_add_js($path . '/jquery.insertAtCaret.js');
  drupal_add_js($path . '/macromaker.js');
  drupal_add_css($path . '/macromaker.css');
}

/**
 * Implementation of hook_theme().
 */
function macromaker_theme() {
  theme = array();
}


```



A screenshot of a Windows File Explorer window. The current directory is 'macromaker'. Inside the folder, there are five files: 'jquery.insertAtCaret.js', 'macromaker.css', 'macromaker.info', 'macromaker.js', and 'macromaker.module'. The 'macromaker.module' file is highlighted with a gray background.