|  |
| --- |
| /\*Form styles\*/.styled {font: 15px Arial, sans-serif;width: 422px;margin: 20px auto;background: url(images/bg\_form.png) no-repeat 0 0;padding-top: 20px;}.styled fieldset {background: url(images/bg\_form.png) no-repeat 0 100%;padding: 0 25px 20px 25px;position: relative;} /\* Form rows \*/.styled fieldset li.form-row {margin-bottom: 5px;padding: 3px 0;clear: both;position: relative; /\* Required as error messages will be absolutely positioned \*/}.styled label {display: block;font-weight: bold;float: left;width: 100px;line-height: 24px;padding-top: 4px;color: #555;}.styled fieldset li.button-row {margin-bottom: 0;padding: 5px 0 0;text-align: right;} /\* Text input styles \*//\* Default \*/.styled input.text-input {height: 22px;width: 254px;padding: 5px 8px;background: url(images/bg\_input.png) no-repeat 0 0;border: none;font: normal 15px Arial, sans-serif;color: #333;line-height: 1em;} |

**The Form Validation CSS**

The following CSS rules are for styling and setting up the form error messages:

[?](http://www.designchemical.com/blog/index.php/jquery/jquery-animated-error-messages-form-validation/)

|  |  |
| --- | --- |
| 123456789101112131415161718192021 | /\* Form Validation \*//\* CSS code to create the error messages \*/.styled span.error {font: bold 11px Arial, sans-serif;line-height: 1em;color:#fff;text-shadow: 1px 1px 1px #000;background: url(images/arrow\_error.png) no-repeat 0 center;border-right: 1px solid #6c0202;height: 11px;padding: 8px 15px 11px 20px;display: none; /\* Required as we will use jQuery to fade in the error message \*/position: absolute; /\* Required \*/top: 3px;right: 0; /\* Required - initially set all error messages to the right of the form row \*/} /\* Change the input field styles when an error message is present \*/.styled fieldset li.error input.text-input {background-position: 0 -64px;} |

**The JQuery Code**

The code below includes comments explaining each function in more detail:

[?](http://www.designchemical.com/blog/index.php/jquery/jquery-animated-error-messages-form-validation/)

|  |  |
| --- | --- |
| 1234567891011121314151617181920212223242526272829303132333435363738394041424344454647484950515253545556575859606162636465666768697071727374757677787980 | // Form validation and submit when button is clicked    $('.btn-submit').click(function(e){         // Declare the function variables:        // Parent form, form URL, email regex and the error HTML        var $formId = $(this).parents('form');        var formAction = $formId.attr('action');        var emailReg = /^([\w-\.]+@([\w-]+\.)+[\w-]{2,4})?$/;        var $error = $('<span class="error"></span>');         // Prepare the form for validation - remove previous errors        $('li',$formId).removeClass('error');        $('span.error').remove();         // Validate all inputs with the class "required"        $('.required',$formId).each(function(){            var inputVal = $(this).val();            var $parentTag = $(this).parent();            if(inputVal == ''){                $parentTag.addClass('error').append($error.clone().text('Required Field'));            }             // Run the email validation using the regex for those input items also having class "email"            if($(this).hasClass('email') == true){                if(!emailReg.test(inputVal)){                    $parentTag.addClass('error').append($error.clone().text('Enter valid email'));                }            }             // Check passwords match for inputs with class "password"            if($(this).hasClass('password') == true){                var password1 = $('#password-1').val();                var password2 = $('#password-2').val();                if(password1 != password2){                $parentTag.addClass('error').append($error.clone().text('Passwords must match'));                }            }        });         // All validation complete - Check if any errors exist        // If has errors        if ($('span.error').length > 0) {             $('span.error').each(function(){                 // Set the distance for the error animation                var distance = 5;                 // Get the error dimensions                var width = $(this).outerWidth();                 // Calculate starting position                var start = width + distance;                 // Set the initial CSS                $(this).show().css({                    display: 'block',                    opacity: 0,                    right: -start+'px'                })                // Animate the error message                .animate({                    right: -width+'px',                    opacity: 1                }, 'slow');             });        } else {            $formId.submit();        }        // Prevent form submission            e.preventDefault();    });     // Fade out error message when input field gains focus    $('.required').focus(function(){        var $parent = $(this).parent();        $parent.removeClass('error');        $('span.error',$parent).fadeOut();    }); |

The comments are fairly self explanatory. So to briefly summarise the jQuery code:

1. When the submit button is clicked the validation process starts by declaring all function variables, including the error message span tag
2. Clean up the form by removing any errors span tags plus the error class on the form rows from the previous click so we are starting with a clean slate

Loop through all input fields with class “required” and first check if they contain text

1. Validate input fields with class “email” using a standard regex pattern for email address formats
2. Get the 2 password field values and check if they match
3. If any of our checks returns an error indentify that form row by adding class=”error”, clone the error span tag, append to the form row and insert the relevant error message text
4. Position each error message span tag to the right of the form row based on the total width of the message
5. Fade in each error message and alert the user by animating 5px to the left
6. Fade out and remove the errors when the user clicks on the text input field