Blog

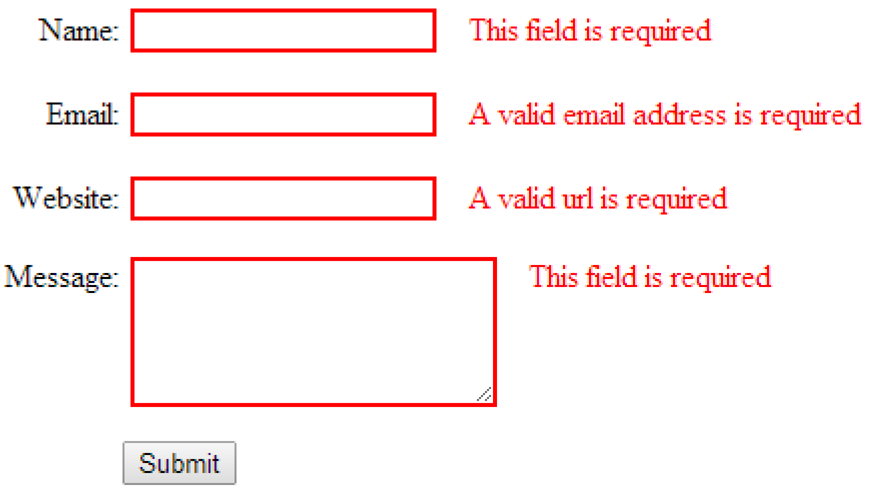
Home (/) / Blog (/blog/) / How to Validate Form Fields Using jQuery

How to Validate Form Fields Using jQuery (/blog/validate-contact-form-jquery)

March 19, 2014 Chris

form (/blog/tag/form) tutorial (/blog/tag/tutorial) validation (/blog/tag/validation) jquery (/blog/tag/jquery)

Update! FormDen can now automatically validate your form fields fields for you. Learn how! (/docs/processing-guide/)



Name: This field is required

Email: A valid email address is required

Website: A valid url is required

Message: This field is required

Show warnings when fields aren't entered correctly

People make mistakes when they fill out forms. By validating form responses before accepting them, we can alert users to their errors before they submit the form. In this way, client side form validation can vastly improve the user experience. This article will explain some basics about form validation and then illustrate how to validate form fields using jQuery.

Use JavaScript Validation to improve UX

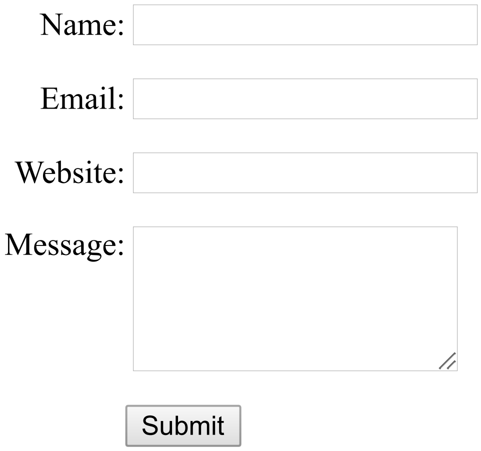
There are two types of validation: client side validation and server side validation. In client side validation the data is validated in the browser using JavaScript before submitting the data to the server. For example, suppose your user submits the form without entering a mandatory field. If you validate the form using jQuery, you can notice this and alert the user to their mistake instead of submitting the form. JavaScript form validation is a great way to help your users avoid mistakes when filling out a form

Don't use JavaScript Validation for Security!

Alternatively, if you want to protect your server from malicious users, then you should use server side validation because JavaScript can be easily bypassed. For example, suppose you wanted to validate a credit card. All credit cards are 16 digits and the 16th digit can be calculated based on the first 15 numbers. This means you can use JavaScript to check for, and alert the user to, credit card typos. However, to see whether a credit card has sufficient funds requires communicating with the credit card company and that can only be done using a server side programming language. Similarly, form data is typically saved in a database on the server. Because your database contains sensitive information, hackers may attempt to use your form to submit malicious commands to your backend database. If you are using a MySQL database, this is called MySQL Injection (http://en.wikipedia.org/wiki/SQL_injection). To avoid this problem, you must validate form submissions using your server. You cannot rely on JavaScript to avoid this security problem.

Demo

Try out our demo below to test our validation!



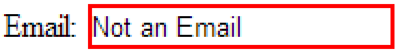
You can also download this demo and check out the source as we go through this tutorial.

Download Demo Source (/static/assets/demos/jquery_validation/validation_demo.zip)

Our form contains two types of validation.

Real-Time Validation

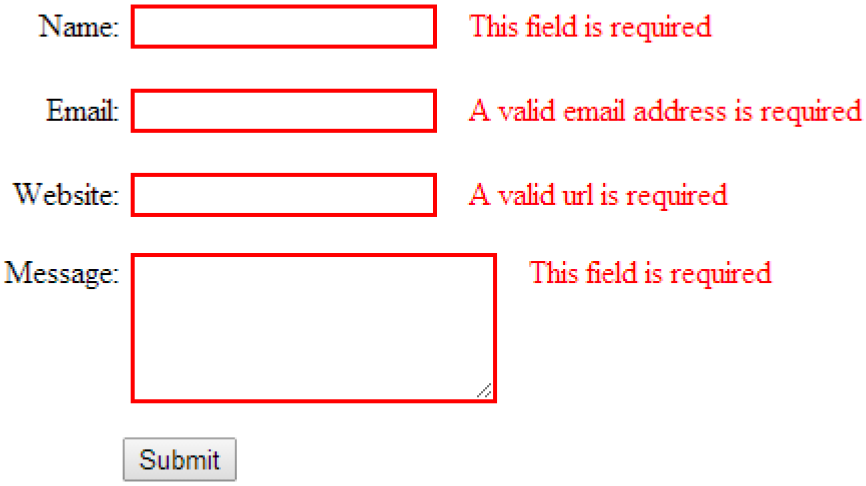
Validate each field as it is being entered. Valid inputs will turn green while invalid inputs will turn red.



Email: Not an Email

Submission Validation

When the submit button is pushed, jQuery will check whether all fields are valid. If any fields are not valid, the form will not be submitted and the user will be informed with error messages for the fields that are causing problems.



Name: This field is required

Email: A valid email address is required

Website: A valid url is required

Message: This field is required

In this tutorial, we will check that:

- The name field has been filled out.
- The email address field looks like an email address.
- The webpage field looks like a url.
- The message field has been filled out.

To get started, we'll create an HTML page that include a form with the id "contact". Each input will be contained in a div. It should contain an label, an input (or textarea), and a span that will contain the error message. The form in HTML will look like this:

```
<form id="contact" method="post" action="">
    <!-- Name -->
    <div>
        <label for="contact_name">Name:</label>
        <input type="text" id="contact_name" name="name"></input>
        <span class="error">This field is required</span>
    </div>
    <!-- Email -->
    <div>
        <label for="contact_email">Email:</label>
        <input type="email" id="contact_email" name="email"></input>
        <span class="error">A valid email address is required</span>
    </div>
    <!--Website -->
    <div>
        <label for="contact_website">Website:</label>
        <input type="url" id="contact_website" name="website"></input>
        <span class="error">A valid url is required</span>
    </div>
    <!-- Message -->
    <div>
        <label for="contact_message">Message:</label>
        <textarea id="contact_message" name="message"></textarea>
        <span class="error">This field is required</span>
    </div>
    <!-- Submit Button -->
    <div id="contact_submit">
        <button type="submit">Submit</button>
    </div>
</form>
```

Note that the <label> tag improves usability. Clicking on the label will move the mouse cursor to the input with the id specified in the for attribute. The name of each input (or textarea) will be passed to the server to identify the contents of the form. Finally, notice that instead of using "text" as the input type for the email and url fields, we use "email" and "url". This will buy us free validation from browsers that support HTML5 even if JavaScript is turned off. However, if JavaScript is turned on, our validation will solve any problems before the browser sees them.

We apply a little CSS to align the labels and space out the fields:

```
#contact label{
    display: inline-block;
    width: 100px;
    text-align: right;
}
#contact_submit{
    padding-left: 100px;
}
#contact div{
    margin-top: 1em;
}
textarea{
    vertical-align: top;
    height: 5em;
}
```

The error messages should only be shown when a field has been entered incorrectly. Therefore, we'll hide them by default. After the form is submitted, we can reveal the error message by changing the class name with JavaScript.

```
.error{
    display: none;
    margin-left: 10px;
}

.error_show{
    color: red;
    margin-left: 10px;
}
```

These error message will only be displayed after the form is submitted. We also want to implement real-time validation where the color of the input will change based on the validity of the data that has been entered. If the data is valid the input box should be green. If the data is invalid, the input box should be red. Therefore, we'll create CSS classes to represent these possible outcomes.

```
input.invalid, textarea.invalid{
    border: 2px solid red;
}

input.valid, textarea.valid{
    border: 2px solid green;
}
```

Set-up jQuery

Now that we've created a working form, we need to add some jQuery and JavaScript to validate fields and change the CSS classes that were created earlier. If you haven't already, make sure that you've included the jQuery library on your page:

```
<script type="text/javascript" src="jquery.min.js"></script>
```

Wrap all of the subsequent jQuery code with:

```
$(document).ready( function () {
    //jQuery code goes here
});
```

so that our scripts are loaded after the page has finished loading.

Use jQuery to Validate in Real-time

We can detect that a user has started entering their name into the name input with:

```
$('#contact_name').on('input', function () {
});
```

Since the name field is required, we simply need to check whether a value for the input exists. If it exists, we'll apply the valid class and remove any invalid class. This will place a green border around the input box. The full code looks like this:

```
// Name can't be blank
$('#contact_name').on('input', function () {
     var input=$( this );
     var is_name=input.val();
     if (is_name){input.removeClass("invalid").addClass("valid");}
     else {input.removeClass("valid").addClass("invalid");}
});
```

The same approach can be used to validate the email address. The only difference is that now we want to check whether the email address looks like an email address. We know that email addresses should include an @ symbol and end with a feasible domain name. To test this, we can use regular expressions. Regular expressions are a programming language for parsing text. Fully understanding them is outside of the scope of this tutorial. However, the basic idea is to specify a pattern and test that pattern against a string of characters. If the test passes, the string is an email address. If it fails, the string is not an email address. The class is adjusted according. All of this is achieved with the following code:

```
// Email must be an email
$('#contact_email').on('input',  function () {
     var  input=$( this );
     var  re = /^[a-zA-Z0-9.!#$%&'*+/=?^_`{|}~-]+@[a-zA-Z0-9-]+(?:\.[a-zA-Z0-9-]+)*$
     var  is_email=re.test(input.val());
     if (is_email){input.removeClass("invalid").addClass("valid");}
     else {input.removeClass("valid").addClass("invalid");}
});
```

A symmetrical approach is used to validate a url. The main difference is that the pattern to identify a url is different from the pattern to identify a website. Also, a valid url must start with http:// but many people omit this and just use www. To help them out, the code will automatically replace www. with http://www (http://www). The validation code is as follows:

```
// Website must be a website
$('#contact_website').on('input',  function () {
     var  input=$( this );
     if  (input.val().substring(0,4)=='www.'){input.val('http://www (http://www).'+i
     var  re = /(http|ftp|https):\/\/[\w-]+(\.[\w-]+)+([\w.,@?^=%&:\/~+#-]*[\w@?^=%&
     var  is_url=re.test(input.val());
     if (is_url){input.removeClass("invalid").addClass("valid");}
     else {input.removeClass("valid").addClass("invalid");}
});
```

Finally, the message validation only requires that something be entered. The code is nearly identical to the name validation:

```
// Message can't be blank
$('#contact_message').keyup( function (event) {
     var  input=$( this );
     var  message=$( this ).val();
    console.log(message);
     if (message){input.removeClass("invalid").addClass("valid");}
     else {input.removeClass("valid").addClass("invalid");}
});
```

Use jQuery to Validate after Submission

Hopefully, your users should be able to use the real-time validation to see whether they have made any errors. However, it's possible they don't notice the red box or understand why their input is not validating. Therefore, we'll double check that all the fields are valid when the submit button is pushed. If a field is not valid, we'll display the error message.

We can capture the submit button being clicked with:

```
$("#contact_submit button").click( function (event){
});
```

Then, we get an array of all of the data that has been submitted. We'll loop through every input in the form. If it does not contain the CSS class "valid", we know that the real-time validation script has found an error in an input. Therefore, we'll change the class from "error" (which is hidden), to "error_show" to display the error message explicitly describing what went wrong. In jQuery, this looks like:

```
// After Form Submitted Validation
$("#contact_submit button").click( function (event){
     var  form_data=$("#contact").serializeArray();
     var  error_free= true ;
     for  ( var  input  in  form_data){
         var  element=$("#contact_"+form_data[input]['name']);
         var  valid=element.hasClass("valid");
         var  error_element=$("span", element.parent());
         if  (!valid){error_element.removeClass("error").addClass("error_show"); err
         else {error_element.removeClass("error_show").addClass("error");}
    }
     if  (!error_free){
        event.preventDefault();
    }
     else {
        alert('No errors: Form will be submitted');
    }
});
```

If there are any problems, we'll prevent the form from submitting. Otherwise, it will submit itself. In this example, we're not dealing with any server side processing. So, we'll simply issue a JavaScript alert to say that everything worked.

You may notice that it would be very easy for someone to get around our validation. They could just change the class on one of the inputs from invalid to valid. This illustrates that JavaScript validation is inherently insecure. Don't rely on it for security. However, it can vastly improve the user experience.

Let us process your forms...

We offer free form processing! POST your forms to our server. We'll validate fields, filter SPAM, email responses and more.

Learn More! (/)

34 Comments **FormDen.com** 1 **Login**

♡ **Recommend** 18 **Share** **Sort by Best**

Join the discussion…

LOG IN WITH

OR SIGN UP WITH DISQUS ?

Name

Prasanth Sunderasan • 10 months ago

I am a beginner for web designing! would you help me with some basic codes for form validation using javascipt or jquery??

1 ∧ | ∨ • Reply • Share ›

anu reena • 2 months ago

Give code for image validation with this method

∧ | ∨ • Reply • Share ›

anu reena • 2 months ago

Reply me fast

∧ | ∨ • Reply • Share ›

anu reena • 2 months ago

I want do image validation this method

∧ | ∨ • Reply • Share ›

Dian Andria • 4 months ago

thank you !

⌃ | ⌄ • Reply • Share ›

Ardiyan Ramadhan • 4 months ago

thank you so much, so helpful

⌃ | ⌄ • Reply • Share ›

Chathura Buddhika • 10 months ago

thank you so much...

⌃ | ⌄ • Reply • Share ›

Piterson Rns Ledrole • 10 months ago

Thank you! for thar.

⌃ | ⌄ • Reply • Share ›

rob • a year ago

Hi Chris,

Thanks for the validation, as a beginner coder this has been the easiest form to understand so far.. I just have two questions:

1. what would be the validation if the fields needed to contain a value that is greater than two characters? more specifically, for the phone number and email..

Any feedback would be greatly appreciated

Rob

⌃ | ⌄ • Reply • Share ›

Chris Mod ➦ rob • a year ago

Hi Rob,
Glad you found the tutorial useful. Here is a regular expression to check whether a string is longer than 2 characters:
var re = /...+/;
(use the same approach as the email validation, except with the different regular expression.

⌃ | ⌄ • Reply • Share ›

rob ➦ Chris • a year ago

Thanks Chris, much appreciated!

⌃ | ⌄ • Reply • Share ›

Rob ➦ rob • a year ago

Hi Chris,
Im currently studying web design/development and I'm really struggling with jQuery..my current assessment requires me to create a form (with required

fields etc) so your tutorial has helped me heaps, and i really appreciate your last reply..I was wondering if i could ask your advice in adding a radio menu and a drop-down menu in the form - i can email you what i've already written if that helps? any feedback would be greatly appreciated - thanks Chris - Rob

^ | v • Reply • Share ›

Rob ➦ Rob • a year ago

I used the following code for the radio menu:

HTML:

```
<div>
<label for="contact_radio" <="" label="">
<input type="radio" id="contact_radio" name="radio">Keen to join Oaktown FC</input>
<label for="contact_radio" <="" label="">
<input type="radio" id="contact_radio" name="radio">Already a member</input>
Select one
</div>
```

jQuery:

```
$('input[name="radio"]').on('change', function() {
var selected_input=$('input[name="radio"]:checked');
var is_myradio=re.test(input.val());
if(is_myradio){input.removeClass("invalid").addClass("valid");}
else{input.removeClass("valid").addClass("invalid");}
});
```

The form submits successfully if i don't check any of the boxes. If i check an option and then click submit, the 'Select one' error will show, so it's doing the opposite of what i need it to; Any feedback would be greatly appreciated mate - thanks:)

^ | v • Reply • Share ›

Muhammad Wajid • a year ago

how do i force a user to not use contact no and url in a text area if he uses then not submit the form

^ | v • Reply • Share ›

Wesley • 2 years ago

Hi
Easiest validation yet - thanks.
Just wanted to know how does one validate contact numbers.
Also when I submit my form the "message" does not come through on the email.
I changed it to "comments" but then the form doesn't work.

^ | v • Reply • Share ›

• Reply • Share ›

Chris Mod ➦ Wesley • 2 years ago

Validation for phone numbers is tricky because they can be formatted in so many ways. It's best to just to make them required. If you are having problems with our form processing, please email support@formden.com with your username and form name.

• Reply • Share ›

guest ➦ Chris • a year ago

add a $("x").mask(000-000-0000)

so that the field comes pre formated and the just follow the guide you set out

• Reply • Share ›

Pooja Dhuri • 2 years ago

Just a question, if we have to clear the span error msg(This field is required) as when the user starts writing, what will be the possible way to do it?

• Reply • Share ›

Hongxuan Choo • 2 years ago

What if you want a radio button validation?

• Reply • Share ›

Chris Mod ➦ Hongxuan Choo • 2 years ago

You would do something like this (where your radio is named myRadio):

```
$('input[name="myRadio"]').on('change', function() {
var selected_input=$('input[name="myRadio"]:checked');
//follow the same pattern in the tutorial
});
```

• Reply • Share ›

sf_79 • 2 years ago

Hello, Thanks for the validation, it's very useful.

I just have a problem when the form field
1. is autocompleted : the autocompleted field is considered as blank;
2. got the value from the session data (you validate for a first time the form, keep the data in the php_session variables, and use it when the user come back on the form page) : the input value comes from the session and is showed in the form, but is considered as blank on the validation.

Could you help me?
Thanks in advance,
sf

• Reply • Share ›

Chris Mod ➦ sf_79 • 2 years ago

Hi sf,

Good question. In the current form, the script runs validation on each field when it is selected by the user. To account for dynamic prepopulated fields, you could simply add the following to trigger validation for each field:

```
$('#contact_name').trigger("input");
$('#contact_email').trigger("input");
$('#contact_website').trigger("input");
$('#contact_message').trigger("keyup");
```

-Chris

⌃ | ⌄ • Reply • Share ›

sf_79 ➦ Chris • 2 years ago

Hi Chris,
I added this after the lignes

```
$("#contact_submit button").click(function(event){
```

It work's perfect.

Thanks for the help, Seb

⌃ | ⌄ • Reply • Share ›

Abdex • 2 years ago

thank you very much !!!
I liked your Demo :)

⌃ | ⌄ • Reply • Share ›

pk • 2 years ago

thank you, this information is very useful to me.

⌃ | ⌄ • Reply • Share ›

AD S • 2 years ago

Thanks!

Just a quick question: how to I exclude a field (eg: make input optional)?

I tried to add $('#optional').addClass('valid'); for those fields but it still didnt work.

EDIT: I found out this was due to including Recaptcha in the form. Added:

```
if(element.selector == "#contact_g-recaptcha-response"){
//DO NOTHING
}
else {
if (!valid){error_element.removeClass("error").addClass("error_show"); error_free=false;
console.log(element); }
else{error_element.removeClass("error_show").addClass("error");}
}
```

^ | v • Reply • Share ›

acasari • 2 years ago

Howdy,

How would you do to insert validation errors under the <input> instead of placing them next to it? In your example the message is hard coded in HTML, but in my scenario I would need to insert them dinamically. Thanks.

^ | v • Reply • Share ›

Chris Mod ➦ acasari • 2 years ago

This is possible but will require some changes to the code. You'll need to use CSS to align the errors where you'd like them (using things like float: left; etc.) Then, instead of using jQuery to add and remove classes, you'll need to use jQuery to add/remove text using .html():
http://api.jquery.com/html/

If you're trying to create a beautiful form from scratch, you might consider using a framework like Bootstrap. We have a free Bootstrap form builder:
https://formden.com/form-bu...
that can be paired with our form processing service to dynamically add error messages.

^ | v • Reply • Share ›

acasari ➦ Chris • 2 years ago

Thanks for your help.

^ | v • Reply • Share ›

Karesz Pola • 2 years ago

Hi Chris,
thanks for your reply.
The demo work's fine too me.I changed the id's.But the submit doesn't work only. I have a bootstrap form and severeal js inside.I think,it's a js affect maybe.

^ | v • Reply • Share ›

Karesz Pola • 2 years ago

HI there,
the validation work's fine.
But if all fields filled correctly,the submit button doesn't call the page given in action="".
Why
Thanks
Kari

^ | v • Reply • Share ›

Chris Mod ➦ Karesz Pola • 2 years ago

Hi Kari,

I just downloaded the demo and verified that it is working correctly. If all the fields are entered correctly, they should be surrounded by a green border and you'll see a JavaScript alert that says: "No errors: Form will be submitted" then the page will go to the url given in action. The attached screenshot is from the relevant part of the code. If there are any errors, the form will not be submitted. That should help you troubleshoot.

• Reply • Share ›

geek Adam • 3 years ago

thank you

• Reply • Share ›

Bjornen Nilsson • 4 months ago

Hi and thank you!

This is really awesome. One question:

How do make all of this into a function for my theme functions.php so that I can apply this to a specific Contact Form 7 form-id?

Thanks!

• Reply • Share ›

ALSO ON **FORMDEN.COM**

How to Isolate Bootstrap CSS to Avoid Conflicts - Blog | Formden.com

20 comments • 3 years ago

Tutorial: Add a Date Picker to a Bootstrap Form | Formden.com

45 comments • 2 years ago

Blog Author

Chris Youderian (http://twitter.com/cyouderian), the founder of FormDen.

Recent Posts

- Guide: Bootstrap Forms by Example (/bootstrap-form-guide/)
- How to Animate a Bootstrap Form with jQuery (/blog/animate)
- How to Customize a Field's Focus Color in a Bootstrap Form (/blog/input-color)
- How to Add a Date Picker to a Bootstrap Form (/blog/date-picker)
- How to Add Icons to Your Bootstrap Form (/blog/addon-icon)
- How to Create Conditional Form Fields with jQuery (/blog/conditional-form-field)
- How to Email a Form Without Using PHP (/blog/process-no-php)
- Tutorial: Bootstrap Form Builder (/blog/tutorial-bootstrap-form-builder)
- How to Isolate Bootstrap CSS to Avoid Conflicts (/blog/isolate-bootstrap)
- Cable companies bundle ESPN and HGTV, should you bundle your products too? (/blog/bundling)
- Product Pricing: Increase your Revenue by Segmenting your Customers (/blog/price-discrimination)
- How to Validate Form Fields Using jQuery (/blog/validate-contact-form-jquery)

Tags

tutorial (/blog/tag/tutorial) form processing (/blog/tag/form-processing) icon (/blog/tag/icon)

addon (/blog/tag/addon) drop shadow (/blog/tag/drop-shadow) validation (/blog/tag/validation)

less (/blog/tag/less) price discrimination (/blog/tag/price-discrimination) date picker (/blog/tag/date-picker)

field focus (/blog/tag/field-focus) forms (/blog/tag/forms) php (/blog/tag/php)

conditional field (/blog/tag/conditional-field) validate (/blog/tag/validate) pricing (/blog/tag/pricing)

bootstrap (/blog/tag/bootstrap) form (/blog/tag/form) css (/blog/tag/css) animate (/blog/tag/animate)

economics (/blog/tag/economics) form builder (/blog/tag/form-builder) jquery (/blog/tag/jquery)

html (/blog/tag/html) bundling (/blog/tag/bundling)

RSS Feed

(/blog/feed/) Subscribe to our RSS feed (/blog/feed/)