

JavaScript Form Validation : quick and easy!

 [HTML Forms](#)  [download, javascript form validation, reference, validation](#)

Using client side JavaScript is an efficient way to validate the user input in web forms. When there are many fields in the form, the JavaScript validation becomes too complex.

The JavaScript class presented here makes the form validations many times easier.

Contents

1. [Download the JavaScript form validation script](#)
2. [Using the form validation script](#)
3. [Adding a custom validation](#)
4. [Table of Validation Descriptors](#)
5. [Showing the form validation errors next to the element](#)
6. ['Conditional' form validations](#)
7. [Form validation without coding!](#)

How to add JavaScript Form Validation quickly

First, [download the JavaScript form validation script here](#).

The zip file contains the javascript file, examples.

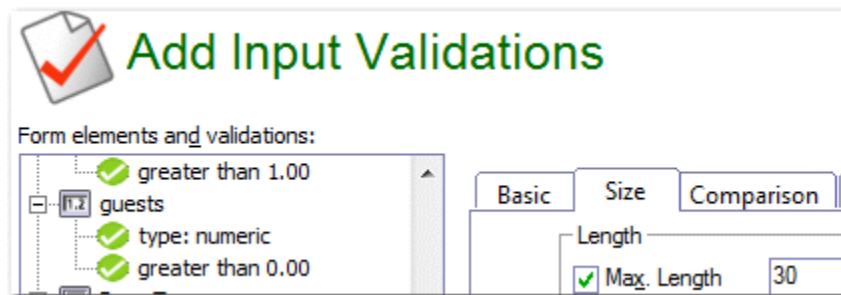
The script has a catalog of almost all the common validation types built-in.

The idea is to create a set of “validation descriptors” associated with each element in a form. The “validation descriptor” is nothing but a string specifying the type of validation to be performed.

Each field in the form can have zero one or more validations. For example, you can have an input field that should not be empty, should be less than 25 chars and should be alpha-numeric.

In other words, in order to validate a field, you just associate a set of validation descriptors for each input field in the form.

Just Choose Validations!



[Simfatic Forms](#) is a feature-rich **web form maker**.

You just have to choose the validations. [More info & downloads](#)

Using the form validation script

1. Include gen_validatorv4.js in your html file just before closing the HEAD tag

```
<script src="gen_validatorv4.js" type="text/javascript"></script>
</head>
```

2. Just after defining your form, create a Validator() object passing the name of the form

```
<form id='myform' action="">
  <!--Your input fields go here -->
</form>
```

```
<script type="text/javascript">
    var frmvalidator  = new Validator("myform");
                                //where myform is the name/id of your form
```

3. Now, add the validations required

```
frmvalidator.addValidation("FirstName","req","Please enter your
First Name");
```

The format of the addValidation() function is:

```
frmvalidator.addValidation(Field Name, Validation Descriptor, Error
String);
```

See below for the [complete list of validation descriptors](#). The third parameter (Error string) is optional.

You can add any number of validations to a field.

```
frmvalidator.addValidation("FirstName","req","Please enter your
First Name");
frmvalidator.addValidation("FirstName","maxlen=40",
                                "Max length for FirstName
is 40");
```

Example

Here is a complete example:

```
<form action="" id="myform" >
<p>
    <label for='FirstName'>First Name:</label>
    <input type="text" id="FirstName" name="FirstName" />
</p>
<p>
    <label for='LastName'>Last Name:</label>
    <input type="text" id="LastName" name="LastName" />
</p>
<p>
    <label for='EMail'>EMail:</label>
    <input type="text" id="EMail" name="EMail" />
</p>
<p>
    <label for='Phone'>Phone:</label>
    <input type="text" id="Phone" name="Phone" />
</p>
<p>
    <label for='Address'>Address:</label>
    <textarea cols="20" rows="5" id="Address" name="Address">
</textarea>
</p>
```

```

<p>
  <label for='Country'>Country:</label>
  <select id="Country" name="Country">
    <option value="000" selected="selected">[choose yours]</option>
    <option value="008">Albania</option>
    <option value="012">Algeria</option>
    <option value="016">American Samoa</option>
    <option value="020">Andorra</option>
    <option value="024">Angola</option>
    <option value="660">Anguilla</option>
    <option value="010">Antarctica</option>
    <option value="028">Antigua And Barbuda</option>
    <option value="032">Argentina</option>
    <option value="051">Armenia</option>
    <option value="533">Aruba</option>
  </select>
</p>
<p>
  <input type="submit" name="submit" value="Submit">
</p>
</form>

<script type="text/javascript">
  var frmvalidator = new Validator("myform");
  frmvalidator.addValidation("FirstName","req","Please enter your First
Name");
  frmvalidator.addValidation("FirstName","maxlen=20",
    "Max length for FirstName is 20");

  frmvalidator.addValidation("LastName","req");
  frmvalidator.addValidation("LastName","maxlen=20");

  frmvalidator.addValidation("Email","maxlen=50");
  frmvalidator.addValidation("Email","req");
  frmvalidator.addValidation("Email","email");

  frmvalidator.addValidation("Phone","maxlen=50");
  frmvalidator.addValidation("Phone","numeric");

  frmvalidator.addValidation("Address","maxlen=50");
  frmvalidator.addValidation("Country","dontselect=000");

</script>

```

Some Additional Notes

- ▶ The form validators should be created only after defining the HTML form (only after the </form> tag.)
- ▶ Your form should have a distinguished name. If there are more than one form in the same page, you can add validators for each of them. The names of the forms and the validators should not clash.

- ▶ You can't use the javascript onsubmit event of the form if you are using this validator script. It is because the validator script automatically overrides the onsubmit event. If you want to add a custom validation, see the section below

Adding a custom validation

If you want to add a custom validation, which is not provided by the validation descriptors, you can do so. Here are the steps:

1. Create a javascript function which returns true or false depending on the validation

```
function DoCustomValidation()
{
    var frm = document.forms["myform"];
    if(frm.pwd1.value != frm.pwd2.value)
    {
        sfm_show_error_msg('The Password and verified password does not match!',frm.pwd1);
        return false;
    }
    else
    {
        return true;
    }
}
```

sfm_show_error_msg() function displays the error message in your chosen style. The first parameter is the error message and the second parameter is the input object.

2. Associate the validation function with the validator object.

```
frmvalidator.setAddnlValidationFunction("DoCustomValidation");
```

The custom validation function will be called automatically after other validations.

If you want to do more than one custom validations, you can do all those validations in the same function.

```
function DoCustomValidation()
{
    var frm = document.forms["myform"];
    if(false == DoMyValidationOne())
    {
        sfm_show_error_msg('Validation One Failed!');
        return false;
    }
    else
```

```

    if(false == DoMyValidationTwo())
    {
        sfm_show_error_msg('Validation Two Failed!');
        return false;
    }
    else
    {
        return true;
    }
}

```

where DoMyValidationOne() and DoMyValidationTwo() are custom functions for validation.

Clear All Validations

In some dynamically programmed pages, it may be required to change the validations in the form at run time. For such cases, a function is included which clears all validations in the validator object.

```
frmvalidator.clearAllValidations();
```

This function call clears all validations you set.

Set focus on validation failure

By default, if there is a validation error, the focus is set on the input element having the error. You can disable this behavior by calling:

```
frmvalidator.EnableFocusOnError(false);
```

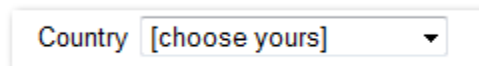
Table of Validation Descriptors

Validation Descriptor Usage

	The field should not be empty.
required or req	Note that this validation is for fields like Textbox and multi-line text box. For 'selections' like drop down and radio group, use an appropriate validation like 'dontselect' or 'selone_radio'.
maxlen=??? or maxlength=???	Limits the length of the input. For example, if the maximum size permitted is 25, give the validation descriptor as "maxlen=25"

Validation Descriptor Usage

minlen=??? or minlength=???	Checks the length of the entered string to the required minimum. Example "minlen=5"
alphanumeric or alnum	The input can contain alphabetic or numeric characters only. (Note that space or punctuation also are not allowed since those characters are not alpha numeric)
alphanumeric_space alnum_s	Allows only alphabetic, numeric and space characters
num numeric	Allow numbers only
alpha alphabetic	Allow only alphabetic characters.
alpha_s alphabetic_space	Allows alphabetic and space characters
email	Validates the field to be a proper email address. (Note, However that the validation can't check whether the email address exists or not)
lt=??? lessthan=???	Verify the data to be less than the value passed. Valid only for numeric fields. Example: if the value should be less than 1000 give validation description as "lt=1000"
gt=??? greaterthan=???	Verify the data to be greater than the value passed. Valid only for numeric fields. Example: if the value should be greater than 10 give validation description as "gt=10"
regexp=???	Match the input with a regular expression. Example: "regexp=^[A-Za-z]{1,20}\$" allow up to 20 alphabetic characters. This validation descriptor is valid only for drop down lists. The drop down select list boxes usually will have one item saying 'Select One' (and that item will be selected by default). The user should select an option other than this 'Select One' item. If the value of this default option is '000', the validation description should be "dontselect=000"
dontselect=??	



```
<SELECT name="Country">  
  <option value="000" selected>[choose yours]
```

Validation Descriptor Usage

dontselectchk=??	<p>This validation descriptor is only for check boxes. The user should not select the given check box. Provide the value of the check box instead of ??</p> <p>For example, dontselectchk=on</p>
shouldselchk=??	<p>This validation descriptor is only for check boxes. The user should select the given check box. Provide the value of the check box instead of ??</p> <p>For example, shouldselchk=on</p>
selone_radio	<p>One of the radio buttons should be selected.</p> <p>Example:</p> <pre>chktestValidator.addValidation("Options", "selone");</pre>

Compare two input elements

egelmnt=???	<p>Compare two input elements. For example: password and confirm password. Replace ??? with the name of the other input element.</p> <p>Example:</p> <pre>frmvalidator.addValidation("confpassword", "egelmnt=password", "The confirmed password is not same as password");</pre> <p>The value should not be equal to the other input element</p> <p>Example:</p>
neelmnt=???	<pre>frmvalidator.addValidation("password", "neelmnt=username", "The password should not be same as username");</pre>
ltelmnt=???	<p>The input should be less than the other input. Give the name of the other input instead of ???</p>
leelmnt=???	<p>The input should be less than or equal to the other input. Give the name of the other input instead of ???</p>
gtelmnt=???	<p>The input should be greater than the other input. Give the name of the other input instead of ???</p>
geelmnt=???	<p>The input should be greater than or equal to the other input. Give the name of the other input instead of ???</p>

Go to the [second part of this post](#) to learn about the advanced features of this validation script.

Learn More

Javascript Form validation Video Tutorial

JavaScript Form Validation Tutorial



- ▶ [A Definitive Guide to Sensible Form Validations](#)
- ▶ [Dreamweaver Form Validation](#)
- ▶ [PHP Form Validation Tutorial](#)



Tired of Building Forms?

Try Simfatic Forms Desktop Form Builder



Related posts:

1. **JavaScript Form Validation Script: More features**
2. **A Form Design Software that is Quick, Easy and Simple.**
3. **How to Submit a Form Using JavaScript**
4. **Using JavaScript to reset or clear a form**
5. **How to set the value of a form field using Javascript PII**
6. **How to set the value of a form field using Javascript PIII**
7. **Simfatic Forms Validation: save your time coding**
8. **Doing real-time calculations in a form using JavaScript**

9. **Using JavaScript to access form objects when there are multiple forms**
10. **Can JavaScript email a form?**
11. **How to get the value of a form element using JavaScript**

❗ Comments on this entry are closed.

↩ **Previous Comments**

bks 



i have solved it look like submit button should have name="submit" and id="submit"
other then that will not submitting value;



bks 



i am using this validator in php form, Problem passing all validation but values are not
"post"

do any one have idea??

Jim 



Almost working perfectly. Cannot for the life of me radio buttons to validate. No error but no
validation when a radio button has to be selected. Any ideas? BTW the demo for radio
buttons doesn't work for me either.

Jim 



Don't worry, worked it out. Had to use frmvalidator instead of chktestValidator!



Evan 

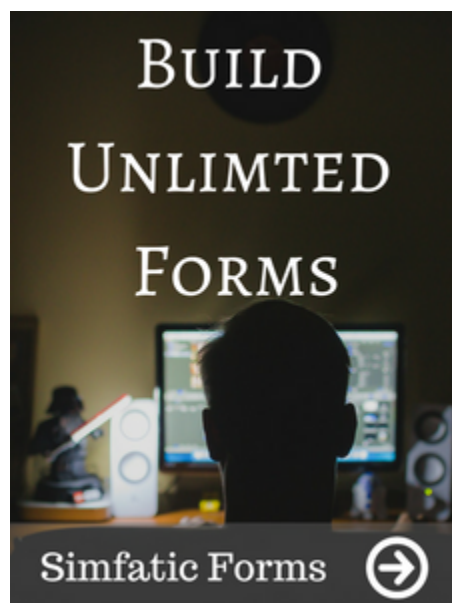


I've tried your validator and it seems to try and validate when the page loads and the input fields are empty. I get an error from the frmvalidator.addValidation() for any fields that do not have default values. Am I doing something incorrectly?

 [Previous Comments](#)

PREVIOUS POST: [USING THE WINDOW.CLOSE METHOD](#)

NEXT POST: [JAVASCRIPT FORM VALIDATION SCRIPT: MORE FEATURES](#)



Categories

[Form Handling](#)

[Form Validation](#)

[HTML Forms](#)

[JavaScript Button](#)

[Javascript Form Handling](#)

[JavaScript Popup Windows](#)

[jQuery](#)

[Tricks & Techniques](#)

[Tutorials](#)

From the Blog

Recent

Popular

Favorites

[jQuery disable button on click to prevent multiple form submits](#)

[Understanding JavaScript Variables](#)

[Using Try...Catch in JavaScript](#)

[A Modern Reintroduction To AJAX](#)

[The Most Indispensable jQuery Form Validation Reference Guide](#)

