# avascript getelementbyid

Have you ever tried to use JavaScript to do some form validation? Did you have any trouble using JavaScript to grab the value of your text field? There's an easy way to access any HTML element, and it's through the use of *id* attributes and the *getElementById* function.

**[Advertise on Tizag.com](http://www.tizag.com/about/advertise.php)**

# javascript document.getelementbyid

If you want to quickly access the value of an HTML input give it an *id* to make your life a lot easier. This small script below will check to see if there is any text in the text field "myText". The argument that *getElementById* requires is the *id* of the HTML element you wish to utilize.

## JavaScript Code:

<script type="text/javascript">

function notEmpty(){

var myTextField = document.getElementById('myText');

if(myTextField.value != "")

alert("You entered: " + myTextField.value)

else

alert("Would you please enter some text?")

}

</script>

<input type='text' id='myText' />

<input type='button' onclick='notEmpty()' value='Form Checker' />

## Display:



*document.getElementById* returned a reference to our HTML element *myText*. We stored this reference into a variable, *myTextField*, and then used the *value*property that all input elements have to use to grab the value the user enters.

There are other ways to accomplish what the above script does, but this is definitely a straight-forward and browser-compatible approach.

# things to remember about getelementbyid

When using the *getElementById* function, you need to remember a few things to ensure that everything goes smoothly. You always need to remember that*getElementById* is a method (or function) of the *document* object. This means you can only access it by using *document.getElementById*.

Also, be sure that you set your HTML elements' id attributes if you want to be able to use this function. Without an id, you'll be dead in the water.

If you want to access the text within a non-input HTML element, then you are going to have to use the *innerHTML* property instead of *value*. The next lesson goes into more detail about the uses of *innerHTML*.

**javascript form validation**

There's nothing more troublesome than receiving orders, guestbook entries, or other form submitted data that are incomplete in some way. You can avoid these headaches once and for all with JavaScript's amazing way to combat bad form data with a technique called "form validation".

**[Advertise on Tizag.com](http://www.tizag.com/about/advertise.php)**

The idea behind JavaScript form validation is to provide a method to check the user entered information before they can even submit it. JavaScript also lets you display helpful alerts to inform the user what information they have entered incorrectly and how they can fix it. In this lesson we will be reviewing some basic form validation, showing you how to check for the following:

* If a text input is empty or not
* If a text input is all numbers
* If a text input is all letters
* If a text input is all alphanumeric characters (numbers & letters)
* If a text input has the correct number of characters in it (useful when restricting the length of a username and/or password)
* If a selection has been made from an HTML select input (the drop down selector)
* If an email address is valid
* How to check all above when the user has completed filling out the form

This lesson is a little long, but knowing how to implement these form validation techniques is definitely worth the effort on your part. Remember to check out Tizag's [HTML forms](http://www.tizag.com/htmlT/forms.php) lesson if you need to brush up on your form knowledge.

**form validation - checking for non-empty**

This has to be the most common type of form validation. You want to be sure that your visitors enter data into the HTML fields you have "required" for a valid submission. Below is the JavaScript code to perform this basic check to see if a given HTML input is empty or not.

**JavaScript Code:**

// If the length of the element's string is 0 then display helper message

function notEmpty(elem, helperMsg){

if(elem.value.length == 0){

alert(helperMsg);

elem.focus(); // set the focus to this input

return false;

}

return true;

}

The function *notEmpty* will check to see that the HTML input that we send it has something in it. *elem* is a HTML text input that we send this function. JavaScriptstrings have built in properties, one of which is the *length* property which returns the length of the string. The chunk of code *elem.value* will grab the string inside the input and by adding on length *elem.value.length* we can see how long the string is.

As long as *elem.value.length* isn't 0 then it's not empty and we return true, otherwise we send an alert to the user with a *helperMsg* to inform them of their error and return false.

**Working Example:**

<script type='text/javascript'>

function notEmpty(elem, helperMsg){

if(elem.value.length == 0){

alert(helperMsg);

elem.focus();

return false;

}

return true;

}

</script>

<form>

Required Field: <input type='text' id='req1'/>

<input type='button'

onclick="notEmpty(document.getElementById('req1'), 'Please Enter a Value')"

value='Check Field' />

</form>

**Display:**

Top of Form

Required Field: 

Bottom of Form

**form validation - checking for all numbers**

If someone is entering a credit card, phone number, zip code, similar information you want to be able to ensure that the input is all numbers. The quickest way to check if an input's string value is all numbers is to use a regular expression /^[0-9]+$/ that will only *match* if the string is all numbers and is at least one character long.

**JavaScript Code:**

// If the element's string matches the regular expression it is all numbers

function isNumeric(elem, helperMsg){

var numericExpression = /^[0-9]+$/;

if(elem.value.match(numericExpression)){

return true;

}else{

alert(helperMsg);

elem.focus();

return false;

}

}

What we're doing here is using JavaScript existing framework to have it do all the hard work for us. Inside each string is a function called *match* that you can use to see if the string matches a certain regular expression. We accessed this function like so: elem.value.match(expressionhere).

We wanted to see if the input's string was all numbers so we made a regular expression to check for numbers [0-9] and stored it as *numericExpression*.

We then used the *match* function with our regular expression. If it is numeric then *match* will return true, making our if statement pass the test and our function*isNumeric* will also return true. However, if the expression fails because there is a letter or other character in our input's string then we'll display our *helperMsg* and return false.

**Working Example:**

<script type='text/javascript'>

function isNumeric(elem, helperMsg){

var numericExpression = /^[0-9]+$/;

if(elem.value.match(numericExpression)){

return true;

}else{

alert(helperMsg);

elem.focus();

return false;

}

}

</script>

<form>

Numbers Only: <input type='text' id='numbers'/>

<input type='button'

onclick="isNumeric(document.getElementById('numbers'), 'Numbers Only Please')"

value='Check Field' />

</form>

**Display:**

Top of Form

Numbers Only: 

Bottom of Form

**form validation - checking for all letters**

This function will be identical to *isNumeric* except for the change to the regular expression we use inside the *match* function. Instead of checking for numbers we will want to check for all letters.

If we wanted to see if a string contained only letters we need to specify an expression that allows for both lowercase and uppercase letters: /^[a-zA-Z]+$/ .

**JavaScript Code:**

// If the element's string matches the regular expression it is all letters

function isAlphabet(elem, helperMsg){

var alphaExp = /^[a-zA-Z]+$/;

if(elem.value.match(alphaExp)){

return true;

}else{

alert(helperMsg);

elem.focus();

return false;

}

}

**Working Example:**

<script type='text/javascript'>

function isAlphabet(elem, helperMsg){

var alphaExp = /^[a-zA-Z]+$/;

if(elem.value.match(alphaExp)){

return true;

}else{

alert(helperMsg);

elem.focus();

return false;

}

}

</script>

<form>

Letters Only: <input type='text' id='letters'/>

<input type='button'

onclick="isAlphabet(document.getElementById('letters'), 'Letters Only Please')"

value='Check Field' />

</form>

**Display:**

Top of Form

Letters Only: 

Bottom of Form

**form validation - checking for numbers and letters**

By combining both the *isAlphabet* and *isNumeric* functions into one we can check to see if a text input contains only letters and numbers.

**JavaScript Code:**

// If the element's string matches the regular expression it is numbers and letters

function isAlphanumeric(elem, helperMsg){

var alphaExp = /^[0-9a-zA-Z]+$/;

if(elem.value.match(alphaExp)){

return true;

}else{

alert(helperMsg);

elem.focus();

return false;

}

}

**form validation - restricting the length**

Being able to restrict the number of characters a user can enter into a field is one of the best ways to prevent bad data. For example, if you know that the zip code field should only be 5 numbers you know that 2 numbers is not sufficient.

Below we have created a *lengthRestriction* function that takes a text field and two numbers. The first number is the minimum number of characters and the second is the maximum number of a characters the input can be. If you just want to specify an exact number then send the same number for both minimum and maximum.

**JavaScript Code:**

function lengthRestriction(elem, min, max){

var uInput = elem.value;

if(uInput.length >= min && uInput.length <= max){

return true;

}else{

alert("Please enter between " +min+ " and " +max+ " characters");

elem.focus();

return false;

}

}

Here's an example of this function for a field that requires 6 to 8 characters for a valid username.

**Working Example:**

<script type='text/javascript'>

function lengthRestriction(elem, min, max){

var uInput = elem.value;

if(uInput.length >= min && uInput.length <= max){

return true;

}else{

alert("Please enter between " +min+ " and " +max+ " characters");

elem.focus();

return false;

}

}

</script>

<form>

Username(6-8 characters): <input type='text' id='restrict'/>

<input type='button'

onclick="lengthRestriction(document.getElementById('restrict'), 6, 8)"

value='Check Field' />

</form>

**Display:**

Top of Form

Username(6-8 characters): 

Bottom of Form

**form validation - selection made**

To be sure that someone has actually selected a choice from an HTML select input you can use a simple trick of making the first option as helpful prompt to the user and a red flag to you for your validation code.

By making the first option of your select input something like "Please Choose" you can spur the user to both make a selection and allow you to check to see if the default option "Please Choose" is still selected when the submit the form.

**JavaScript Code:**

function madeSelection(elem, helperMsg){

if(elem.value == "Please Choose"){

alert(helperMsg);

elem.focus();

return false;

}else{

return true;

}

}

**Working Example:**

<script type='text/javascript'>

function madeSelection(elem, helperMsg){

if(elem.value == "Please Choose"){

alert(helperMsg);

elem.focus();

return false;

}else{

return true;

}

}

</script>

<form>

Selection: <select id='selection'>

<option>Please Choose</option>

<option>CA</option>

<option>WI</option>

<option>XX</option>

</select>

<input type='button'

onclick="madeSelection(document.getElementById('selection'), 'Please Choose Something')"

value='Check Field' />

</form>

**Display:**

Top of Form

Selection:     

Bottom of Form

**form validation - email validation**

And for our grand finale we will be showing you how to check to see if a user's email address is valid. Every email is made up for 5 parts:

1. A combination of letters, numbers, periods, hyphens, plus signs, and/or underscores
2. The at symbol @
3. A combination of letters, numbers, hyphens, and/or periods
4. A period
5. The top level domain (com, net, org, us, gov, ...)

Valid Examples:

* bobby.jo@filltank.net
* jack+jill@hill.com
* the-stand@steven.king.com

Invalid Examples:

* @deleted.net - no characters before the @
* free!dom@bravehe.art - invalid character !
* shoes@need\_shining.com - underscores are not allowed in the domain name

The regular expression to check for all of this is a little overkill and beyond the scope of this tutorial to explain thoroughly. However, test it out and you'll see that it gets the job done.

**JavaScript Code:**

function emailValidator(elem, helperMsg){

var emailExp = /^[\w\-\.\+]+\@[a-zA-Z0-9\.\-]+\.[a-zA-z0-9]{2,4}$/;

if(elem.value.match(emailExp)){

return true;

}else{

alert(helperMsg);

elem.focus();

return false;

}

}

**Working Example:**

<script type='text/javascript'>

function emailValidator(elem, helperMsg){

var emailExp = /^[\w\-\.\+]+\@[a-zA-Z0-9\.\-]+\.[a-zA-z0-9]{2,4}$/;

if(elem.value.match(emailExp)){

return true;

}else{

alert(helperMsg);

elem.focus();

return false;

}

}

</script>

<form>

Email: <input type='text' id='emailer'/>

<input type='button'

onclick="emailValidator1(document.getElementById('emailer'), 'Not a Valid Email')"

value='Check Field' />

</form>

**Display:**

Top of Form

Email: 

Bottom of Form

**validating a form - all at once**

If you've made it this far I commend you, but we're not done yet! The final step is to be able to perform all of these validation steps when the user is ready to submit their data.

Each form has a JavaScript event called *onSubmit* that is triggered when its*submit* button is clicked. If this even returns 0 or false then a form cannot be submitted, and if it returns 1 or true it will always be submitted. Wouldn't it be perfect if we could somehow make an if statement that said "If the form is valid submit it (1) else don't submit it (0)"? Well with a master *formValidator* function we can do just that.

*formValidator* will be somewhat like a list of checks that we want to do before a form is submitted. But before we can decide what we want to check for, we need to have our form!

**HTML Form Code:**

<form onsubmit='return formValidator()' >

First Name: <input type='text' id='firstname' /><br />

Address: <input type='text' id='addr' /><br />

Zip Code: <input type='text' id='zip' /><br />

State: <select id='state'>

<option>Please Choose</option>

<option>AL</option>

<option>CA</option>

<option>TX</option>

<option>WI</option>

</select><br />

Username(6-8 characters): <input type='text' id='username' /><br />

Email: <input type='text' id='email' /><br />

<input type='submit' value='Check Form' /><br />

</form>

That's a lot of data to verify and the first thing we would probably want to check is that each field was at least filled out. To check for completion we will ensure no fields are empty and that the *SELECT* field has a selection. Here are the starting pieces of our master validation function *formValidator*.

**JavaScript Code:**

function formValidator(){

// Make quick references to our fields

var firstname = document.getElementById('firstname');

var addr = document.getElementById('addr');

var zip = document.getElementById('zip');

var state = document.getElementById('state');

var username = document.getElementById('username');

var email = document.getElementById('email');

// Check each input in the order that it appears in the form!

if(isAlphabet(firstname, "Please enter only letters for your name")){

if(isAlphanumeric(addr, "Numbers and Letters Only for Address")){

if(isNumeric(zip, "Please enter a valid zip code")){

if(madeSelection(state, "Please Choose a State")){

if(lengthRestriction(username, 6, 8)){

if(emailValidator(email, "Please enter a valid email address")){

return true;

}

}

}

}

}

}

return false;

}

The first part of this function is where we create easy references to our HTML inputs using the *getElementById* function. These quick references will make our next block of code much easier to read!

The second part uses a bunch of embedded if statements to see whether or not each field has the correct type of data. If every single one of those fields we check validates, then we'll return true and the form will be submitted successfully.

However, if just one of those if statements fails then the *return false* at the end of the function is reached and prevents the form for being submitted.

As you can see this function really does do quite a lot, definitely earning the title of *formValidator*. Notice how this one function references all of the functions we have covered in this lesson. By placing all of these checks in a central location you make your code easier to read and easier to change around in the future.

Now let's put all the necessary and HTML together and try it out!

**all together now**

Below we have taken the HTML form code and the new function *formValidator*and plugged in all the other form validation functions taught in this lesson that are referenced in *formValidator*.

**HTML & JavaScript Code:**

<script type='text/javascript'>

function formValidator(){

// Make quick references to our fields

var firstname = document.getElementById('firstname');

var addr = document.getElementById('addr');

var zip = document.getElementById('zip');

var state = document.getElementById('state');

var username = document.getElementById('username');

var email = document.getElementById('email');

// Check each input in the order that it appears in the form!

if(isAlphabet(firstname, "Please enter only letters for your name")){

if(isAlphanumeric(addr, "Numbers and Letters Only for Address")){

if(isNumeric(zip, "Please enter a valid zip code")){

if(madeSelection(state, "Please Choose a State")){

if(lengthRestriction(username, 6, 8)){

if(emailValidator(email, "Please enter a valid email address")){

return true;

}

}

}

}

}

}

return false;

}

function notEmpty(elem, helperMsg){

if(elem.value.length == 0){

alert(helperMsg);

elem.focus(); // set the focus to this input

return false;

}

return true;

}

function isNumeric(elem, helperMsg){

var numericExpression = /^[0-9]+$/;

if(elem.value.match(numericExpression)){

return true;

}else{

alert(helperMsg);

elem.focus();

return false;

}

}

function isAlphabet(elem, helperMsg){

var alphaExp = /^[a-zA-Z]+$/;

if(elem.value.match(alphaExp)){

return true;

}else{

alert(helperMsg);

elem.focus();

return false;

}

}

function isAlphanumeric(elem, helperMsg){

var alphaExp = /^[0-9a-zA-Z]+$/;

if(elem.value.match(alphaExp)){

return true;

}else{

alert(helperMsg);

elem.focus();

return false;

}

}

function lengthRestriction(elem, min, max){

var uInput = elem.value;

if(uInput.length >= min && uInput.length <= max){

return true;

}else{

alert("Please enter between " +min+ " and " +max+ " characters");

elem.focus();

return false;

}

}

function madeSelection(elem, helperMsg){

if(elem.value == "Please Choose"){

alert(helperMsg);

elem.focus();

return false;

}else{

return true;

}

}

function emailValidator(elem, helperMsg){

var emailExp = /^[\w\-\.\+]+\@[a-zA-Z0-9\.\-]+\.[a-zA-z0-9]{2,4}$/;

if(elem.value.match(emailExp)){

return true;

}else{

alert(helperMsg);

elem.focus();

return false;

}

}

</script>

<form onsubmit='return formValidator()' >

First Name: <input type='text' id='firstname' /><br />

Address: <input type='text' id='addr' /><br />

Zip Code: <input type='text' id='zip' /><br />

State: <select id='state'>

<option>Please Choose</option>

<option>AL</option>

<option>CA</option>

<option>TX</option>

<option>WI</option>

</select><br />

Username(6-8 characters): <input type='text' id='username' /><br />

Email: <input type='text' id='email' /><br />

<input type='submit' value='Check Form' />

</form>

**Display:**

Top of Form

First Name:   
Address:   
Zip Code:   
State:            
Username(6-8 characters):   
Email:   


## Form Validation

In this document we have discussed JavaScript Form Validation using a sample registration form. The tutorial explores JavaScript validation on submit with detail explanation.

Following pictorial shows in which field, what validation we want to impose.

## How would we set those validations

We will create JavaScript functions (one for each input field whose value is to validated) which checks whether a value submitted by user passes the validation.

All those functions are called into another function.

It sets the focus to the input field until the user supplies a valid value.

When user does so, they may proceed and can supply value to the next available field.

The later JavaScript function created is called on onsubmit event of the form.

## HTML Code of the Sample Registration Form

[view plaincopy to clipboardprint?](http://www.w3resource.com/javascript/form/javascript-sample-registration-form-validation.php)

1. <!DOCTYPE html**>**
2. **<html** lang="en"**><head>**
3. **<meta** charset="utf-8"**>**
4. **<title>**JavaScript Form Validation using a sample registration form**</title>**
5. **<meta** name="keywords" content="example, JavaScript Form Validation, Sample registration form" **/>**
6. **<meta** name="description" content="This document is an example of JavaScript Form Validation using a sample registration form. " **/>**
7. **<link** rel='stylesheet' href='js-form-validation.css' type='text/css' **/>**
8. **<script** src="sample-registration-form-validation.js"**></script>**
9. **</head>**
10. **<body** onload="document.registration.userid.focus();"**>**
11. **<h1>**Registration Form**</h1>**
12. **<p>**Use tab keys to move from one input field to the next.**</p>**
13. **<form** name='registration' onSubmit="return formValidation();"**>**
14. **<ul>**
15. **<li><label** for="userid"**>**User id:**</label></li>**
16. **<li><input** type="text" name="userid" size="12" **/></li>**
17. **<li><label** for="passid"**>**Password:**</label></li>**
18. **<li><input** type="password" name="passid" size="12" **/></li>**
19. **<li><label** for="username"**>**Name:**</label></li>**
20. **<li><input** type="text" name="username" size="50" **/></li>**
21. **<li><label** for="address"**>**Address:**</label></li>**
22. **<li><input** type="text" name="address" size="50" **/></li>**
23. **<li><label** for="country"**>**Country:**</label></li>**
24. **<li><select** name="country"**>**
25. **<option** selected="" value="Default"**>**(Please select a country)**</option>**
26. **<option** value="AF"**>**Australia**</option>**
27. **<option** value="AL"**>**Canada**</option>**
28. **<option** value="DZ"**>**India**</option>**
29. **<option** value="AS"**>**Russia**</option>**
30. **<option** value="AD"**>**USA**</option>**
31. **</select></li>**
32. **<li><label** for="zip"**>**ZIP Code:**</label></li>**
33. **<li><input** type="text" name="zip" **/></li>**
34. **<li><label** for="email"**>**Email:**</label></li>**
35. **<li><input** type="text" name="email" size="50" **/></li>**
36. **<li><label** id="gender"**>**Sex:**</label></li>**
37. **<li><input** type="radio" name="msex" value="Male" **/><span>**Male**</span></li>**
38. **<li><input** type="radio" name="fsex" value="Female" **/><span>**Female**</span></li>**
39. **<li><label>**Language:**</label></li>**
40. **<li><input** type="checkbox" name="en" value="en" checked **/><span>**English**</span></li>**
41. **<li><input** type="checkbox" name="nonen" value="noen" **/><span>**Non English**</span></li>**
42. **<li><label** for="desc"**>**About:**</label></li>**
43. **<li><textarea** name="desc" id="desc"**></textarea></li>**
44. **<li><input** type="submit" name="submit" value="Submit" **/></li>**
45. **</ul>**
46. **</form>**
47. **</body>**
48. **</html>**

sample-registration-form-validation.js is the external JavaScript file which contains the JavaScript ocde used to validate the form. js-form-validation.css is the stylesheet containing styles for the form. Notice that for validation, the JavaScript function containing the code to validate is called on the onSubmit event of the form.

For the sake of demonstration, we have taken five countries only. You may add any number of countries in the list.

## CSS Code of the Sample Registration Form

[view plaincopy to clipboardprint?](http://www.w3resource.com/javascript/form/javascript-sample-registration-form-validation.php)

1. h1 {
2. **margin-left**: 70px;
3. }
4. form li {
5. **list-style**: none;
6. **margin-bottom**: 5px;
7. }
9. form ul li label{
10. **float**: left;
11. **clear**: left;
12. **width**: 100px;
13. **text-align**: right;
14. **margin-right**: 10px;
15. **font-family**:Verdana, Arial, Helvetica, sans-serif;
16. **font-size**:14px;
17. }
19. form ul li input, select, span {
20. **float**: left;
21. **margin-bottom**: 10px;
22. }
24. form textarea {
25. **float**: left;
26. **width**: 350px;
27. **height**: 150px;
28. }
30. [type="submit"] {
31. **clear**: left;
32. **margin**: 20px 0 0 230px;
33. **font-size**:18px
34. }
36. p {
37. **margin-left**: 70px;
38. **font-weight**: bold;
39. }

## JavaScript code for validation

## JavaScript function which is called on onSubmit

This function call all other functions used for validation.

[view plaincopy to clipboardprint?](http://www.w3resource.com/javascript/form/javascript-sample-registration-form-validation.php)

1. **function** formValidation()
2. {
3. **var** uid = document.registration.userid;
4. **var** passid = document.registration.passid;
5. **var** uname = document.registration.username;
6. **var** uadd = document.registration.address;
7. **var** ucountry = document.registration.country;
8. **var** uzip = document.registration.zip;
9. **var** uemail = document.registration.email;
10. **var** umsex = document.registration.msex;
11. **var** ufsex = document.registration.fsex; **if**(userid\_validation(uid,5,12))
12. {
13. **if**(passid\_validation(passid,7,12))
14. {
15. **if**(allLetter(uname))
16. {
17. **if**(alphanumeric(uadd))
18. {
19. **if**(countryselect(ucountry))
20. {
21. **if**(allnumeric(uzip))
22. {
23. **if**(ValidateEmail(uemail))
24. {
25. **if**(validsex(umsex,ufsex))
26. {
27. }
28. }
29. }
30. }
31. }
32. }
33. }
34. }
35. **return** **false**;
36. }

## JavaScript function for validating userid

[view plaincopy to clipboardprint?](http://www.w3resource.com/javascript/form/javascript-sample-registration-form-validation.php)

1. **function** userid\_validation(uid,mx,my)
2. {
3. **var** uid\_len = uid.value.length;
4. **if** (uid\_len == 0 || uid\_len >= my || uid\_len < mx)
5. {
6. alert("User Id should not be empty / length be between "+mx+" to "+my);
7. uid.focus();
8. **return** **false**;
9. }
10. **return** **true**;
11. }

The code above checks whether userid input field is provided with a string of length 5 to 12 characters. If not, it displays an alert.

**Flowchart :**

## JavaScript function for validating password

[view plaincopy to clipboardprint?](http://www.w3resource.com/javascript/form/javascript-sample-registration-form-validation.php)

1. **function** passid\_validation(passid,mx,my)
2. {
3. **var** passid\_len = passid.value.length;
4. **if** (passid\_len == 0 ||passid\_len >= my || passid\_len < mx)
5. {
6. alert("Password should not be empty / length be between "+mx+" to "+my);
7. passid.focus();
8. **return** **false**;
9. }
10. **return** **true**;
11. }

The above code used to validate password (it should be of length 7 to 12 characters). If not, it displays an alert.

**Flowchart :**

## JavaScript code for validating user name

[view plaincopy to clipboardprint?](http://www.w3resource.com/javascript/form/javascript-sample-registration-form-validation.php)

1. **function** allLetter(uname)
2. {
3. **var** letters = /^[A-Za-z]+$/;
4. **if**(uname.value.match(letters))
5. {
6. **return** **true**;
7. }
8. **else**
9. {
10. alert('Username must have alphabet characters only');
11. uname.focus();
12. **return** **false**;
13. }
14. }

The code above checks whether user name input field is provided with alphabates characters. If not, it displays an alert.

**Flowchart :**

## JavaScript code for validating user address

[view plaincopy to clipboardprint?](http://www.w3resource.com/javascript/form/javascript-sample-registration-form-validation.php)

1. **function** alphanumeric(uadd)
2. {
3. **var** letters = /^[0-9a-zA-Z]+$/;
4. **if**(uadd.value.match(letters))
5. {
6. **return** **true**;
7. }
8. **else**
9. {
10. alert('User address must have alphanumeric characters only');
11. uadd.focus();
12. **return** **false**;
13. }
14. }

The code above checks whether user address input field is provided with alphanumeric characters. If not, it displays an alert.

**Flowchart :**

## JavaScript code for validating country

[view plaincopy to clipboardprint?](http://www.w3resource.com/javascript/form/javascript-sample-registration-form-validation.php)

1. **function** countryselect(ucountry)
2. {
3. **if**(ucountry.value == "Default")
4. {
5. alert('Select your country from the list');
6. ucountry.focus();
7. **return** **false**;
8. }
9. **else**
10. {
11. **return** **true**;
12. }
13. }

The code above checks whether a country is selected from the given list. If not, then it displays an alert.

**Flowchart :**

## JavaScript code for validating ZIP code

[view plaincopy to clipboardprint?](http://www.w3resource.com/javascript/form/javascript-sample-registration-form-validation.php)

1. **function** allnumeric(uzip)
2. {
3. **var** numbers = /^[0-9]+$/;
4. **if**(uzip.value.match(numbers))
5. {
6. **return** **true**;
7. }
8. **else**
9. {
10. alert('ZIP code must have numeric characters only');
11. uzip.focus();
12. **return** **false**;
13. }
14. }

The code above checks whether a ZIP code of numeric value. If not, it displays an alert.

**Flowchart :**

## JavaScript code for validating email format

[view plaincopy to clipboardprint?](http://www.w3resource.com/javascript/form/javascript-sample-registration-form-validation.php)

1. **function** ValidateEmail(uemail)
2. {
3. **var** mailformat = /^\w+([\.-]?\w+)\*@\w+([\.-]?\w+)\*(\.\w{2,3})+$/;
4. **if**(uemail.value.match(mailformat))
5. {
6. **return** **true**;
7. }
8. **else**
9. {
10. alert("You have entered an invalid email address!");
11. uemail.focus();
12. **return** **false**;
13. }
14. }

The code above checks whether a valid email format is supplied. If not,it displays an alert.

**Flowchart :**

## JavaScript code for validating gender

[view plaincopy to clipboardprint?](http://www.w3resource.com/javascript/form/javascript-sample-registration-form-validation.php)

1. **function** validsex(umsex,ufsex)
2. {
3. x=0;
5. **if**(umsex.checked)
6. {
7. x++;
8. } **if**(ufsex.checked)
9. {
10. x++;
11. }
12. **if**(x==0)
13. {
14. alert('Select Male/Female');
15. umsex.focus();
16. **return** **false**;
17. }
18. **else**
19. {
20. alert('Form Successfully Submitted');
21. window.location.reload()
22. **return** **true**;}
23. }

The code above checks whether a sex is selected. If not, it displays an alert. If Male or Female is selected, it generates an alert saying that the form is successfully submitted and it reloads the form.

**Flowchart :**

Here is the entire JavaScript used for validation of the form.

[view plaincopy to clipboardprint?](http://www.w3resource.com/javascript/form/javascript-sample-registration-form-validation.php)

1. **function** formValidation()
2. {
3. **var** uid = document.registration.userid;
4. **var** passid = document.registration.passid;
5. **var** uname = document.registration.username;
6. **var** uadd = document.registration.address;
7. **var** ucountry = document.registration.country;
8. **var** uzip = document.registration.zip;
9. **var** uemail = document.registration.email;
10. **var** umsex = document.registration.msex;
11. **var** ufsex = document.registration.fsex; **if**(userid\_validation(uid,5,12))
12. {
13. **if**(passid\_validation(passid,7,12))
14. {
15. **if**(allLetter(uname))
16. {
17. **if**(alphanumeric(uadd))
18. {
19. **if**(countryselect(ucountry))
20. {
21. **if**(allnumeric(uzip))
22. {
23. **if**(ValidateEmail(uemail))
24. {
25. **if**(validsex(umsex,ufsex))
26. {
27. }
28. }
29. }
30. }
31. }
32. }
33. }
34. }
35. **return** **false**;
37. } **function** userid\_validation(uid,mx,my)
38. {
39. **var** uid\_len = uid.value.length;
40. **if** (uid\_len == 0 || uid\_len >= my || uid\_len < mx)
41. {
42. alert("User Id should not be empty / length be between "+mx+" to "+my);
43. uid.focus();
44. **return** **false**;
45. }
46. **return** **true**;
47. }
48. **function** passid\_validation(passid,mx,my)
49. {
50. **var** passid\_len = passid.value.length;
51. **if** (passid\_len == 0 ||passid\_len >= my || passid\_len < mx)
52. {
53. alert("Password should not be empty / length be between "+mx+" to "+my);
54. passid.focus();
55. **return** **false**;
56. }
57. **return** **true**;
58. }
59. **function** allLetter(uname)
60. {
61. **var** letters = /^[A-Za-z]+$/;
62. **if**(uname.value.match(letters))
63. {
64. **return** **true**;
65. }
66. **else**
67. {
68. alert('Username must have alphabet characters only');
69. uname.focus();
70. **return** **false**;
71. }
72. }
73. **function** alphanumeric(uadd)
74. {
75. **var** letters = /^[0-9a-zA-Z]+$/;
76. **if**(uadd.value.match(letters))
77. {
78. **return** **true**;
79. }
80. **else**
81. {
82. alert('User address must have alphanumeric characters only');
83. uadd.focus();
84. **return** **false**;
85. }
86. }
87. **function** countryselect(ucountry)
88. {
89. **if**(ucountry.value == "Default")
90. {
91. alert('Select your country from the list');
92. ucountry.focus();
93. **return** **false**;
94. }
95. **else**
96. {
97. **return** **true**;
98. }
99. }
100. **function** allnumeric(uzip)
101. {
102. **var** numbers = /^[0-9]+$/;
103. **if**(uzip.value.match(numbers))
104. {
105. **return** **true**;
106. }
107. **else**
108. {
109. alert('ZIP code must have numeric characters only');
110. uzip.focus();
111. **return** **false**;
112. }
113. }
114. **function** ValidateEmail(uemail)
115. {
116. **var** mailformat = /^\w+([\.-]?\w+)\*@\w+([\.-]?\w+)\*(\.\w{2,3})+$/;
117. **if**(uemail.value.match(mailformat))
118. {
119. **return** **true**;
120. }
121. **else**
122. {
123. alert("You have entered an invalid email address!");
124. uemail.focus();
125. **return** **false**;
126. }
127. } **function** validsex(umsex,ufsex)
128. {
129. x=0;
131. **if**(umsex.checked)
132. {
133. x++;
134. } **if**(ufsex.checked)
135. {
136. x++;
137. }
138. **if**(x==0)
139. {
140. alert('Select Male/Female');
141. umsex.focus();
142. **return** **false**;
143. }
144. **else**
145. {
146. alert('Form Succesfully Submitted');
147. window.location.reload()
148. **return** **true**;
149. }
150. }

<html>  
<head>  
<meta http-equiv="Content-Type" content="text/html; charset=utf-8"/>  
<title>Registration form with validation</title>  
</head>  
<body>  
<center>  
<h1 align="center">Registration form with validation</h1>  
<table border="1">  
<tr>  
<td>  
<form name="registerationform" method="POST" action="welcome.html" onsubmit="return(regvalidate())">  
<table>  
<tr>  
<td>First Name: </td> <td><input type="text" name="fnametxt"/>  
</td>  
</tr>  
<tr>  
<td>Second Name: </td> <td> <input type="text" name="snametxt"/>  
</td>  
</tr>  
<tr>  
<td> User Name:</td> <td><input type="text" name="unametxt"/>  
</td>  
</tr>  
<tr>  
<td>Email Address: </td> <td> <input type="text" name="emailtxt"/>  
</td>  
</tr>  
</tr>  
<tr>  
<td> Password : </td> <td> <input type="password" name="pwdtxt"/>   
</td>  
</tr>  
</tr>  
<tr>  
<td> Confirm : </td> <td> <input type="password" name="cpwdtxt"/>   
</td>  
</tr>  
</table>  
<font color="red"> <div id="une" align="center"> </div> </font>  
<div>  </div>  
<input type="submit" value="Register Now" style="margin-left:  
140px"/>  
</form>  
</td>  
</th>  
</tr>  
</table>  
</tr>  
</table>  
</tr>  
<script type="Text/JavaScript">  
function regvalidate()

{  
if((document.registerationform.fnametxt.value=="")&&(document.registerationform.snametxt.value==""))  
{  
document.getElementById('une').innerHTML = "First name or Second name should not be empty";  
registerationform.fnametxt.focus();  
return(false);  
}

if(document.registerationform.unametxt.value=="")  
{  
document.getElementById('une').innerHTML = "User name field should not be empty";  
registerationform.unametxt.focus();  
return(false);  
}

if(document.registerationform.emailtxt.value=="")  
{  
document.getElementById('une').innerHTML = "Email id requered";  
registerationform.emailtxt.focus();  
return(false);  
}

if(document.registerationform.pwdtxt.value=="")  
{  
document.getElementById('une').innerHTML = "Please type a password";  
registerationform.pwdtxt.focus();  
return(false);  
}

if((document.registerationform.pwdtxt.value) != (document.registerationform.cpwdtxt.value))  
{  
document.getElementById('une').innerHTML = "Password Must be equal";  
registerationform.pwdtxt.value = "";  
registerationform.cpwdtxt.value = "";  
registerationform.pwdtxt.focus();  
return(false);  
}  
else  
{  
return(true);  
}  
}  
</script>  
</td>  
</tr>  
</table>  
</center>  
</body>  
</html>

<!DOCTYPE html PUBLIC "-//W3C//DTD XHTML 1.0 Transitional//EN" "http://www.w3.org/TR/xhtml1/DTD/xhtml1-transitional.dtd">

<html xmlns="http://www.w3.org/1999/xhtml">

<head>

<link rel="stylesheet" type="text/css" href="StyleSheet.css"/></head>

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

<html>

<body>

<div id="form">

<form name="registration" onsubmit="return validate" method="post">

<h3>Registration Form</h3>

<fieldset id="Logon">

<legend>Create Logon</legend>

<!-- add a class to all the input boxes instead. Safer and more backwards-compatible -->

<div class="form-line">

<label for="username">Username \*: </label>

<input type="text" id="Username" name="userid"/></div>

<div class="form-line">

<label for="pwd">Password \*: </label>

<input type="password" id="Password" name="pwd" /></div>

<div class="form-line">

<label for="pwd">Confirm Password \*: </label>

<input type="password" id="Confirm Password" name="confpwd"/></div>

</fieldset>

<script type="text/javascript">

function validate()

{

//registration form function - field variables

var userid = document.form["registration"]["userid"].value;

var pwd = document.form["registration"]["pwd"].value;

var confpwd = document.form["registration"]["confpwd"].value;

var lastname = document.form["registration"]["lastname"].value;

var fname = document.form["registration"]["fname"].value;

var dob = document.form["registration"]["dob"].value;

var email = document.form["registration"]["email"].value;

var confemail = document.form["registration"]["confemail"].value;

var tel = document.form["registration"]["tel"].value;

var add1 = document.form["registration"]["add1"].value;

var add2 = document.form["registration"]["add2"].value;

var twnc = document.form["registration"]["twnc"].value;

var ptc = document.form["registration"]["ptc"].value;

var valid = true;

var username=document.getElementById("userid");

if(username.value=="")

{

alert("Please enter username");

username.focus();

return false;

}

var Password=document.getElementById("pwd");

if(Password.value=="")

{

alert("Please enter Password");

Password.focus();

return false;

}

var rePassword=document.getElementById("confpwd");

if(rePassword.value=="")

{

alert("Please re-enter password");

rePassword.focus();

return false;

}

var dob=document.getElementById("dob");

if(dob.value=="")

{

alert("Please re-enter dob");

dob.focus();

return false;

}

var email=document.getElementById("email");

if(email.value=="")

{

alert("Please re-enter email");

email.focus();

return false;

}

var cemail=document.getElementById("confemail");

if(cemail.value=="")

{

alert("Please re-enter email");

cemail.focus();

return false;

}

var address=document.getElementById("add1");

if(address.value=="")

{

alert("Please re-enter address");

address.focus();

return false;

}

var city=document.getElementById("twnc");

if(city.value=="")

{

alert("Please re-enter town/city");

city.focus();

return false;

}

var post=document.getElementById("ptc");

if(pin.value=="")

{

alert("Please re-enter Postcode");

pin.focus();

return false;

}

}

</script

<html>

<body>

<script>

function validateform(){

var name=document.myform.name.value;

var password=document.myform.password.value;

if (name==null || name==""){

alert("Name can't be blank");

return false;

}else if(password.length<6){

alert("Password must be at least 6 characters long.");

return false;

}

}

</script>

<body>

<form name="myform" method="post" action="http://www.javatpoint.com/javascriptpages/valid.jsp" onsubmit="return validateform()" >

Name: <input type="text" name="name"><br/>

Password: <input type="password" name="password"><br/>

<input type="submit" value="register">

</form>

</body>

</html>

JavaScript Retype Password Validation

<!DOCTYPE html>

<html>

<head>

<script type="text/javascript">

function matchpass(){

var firstpassword=document.f1.password.value;

var secondpassword=document.f1.password2.value;

if(firstpassword==secondpassword){

return true;

}

else{

alert("password must be same!");

return false;

}

}

</script>

</head>

<body>

<form name="f1" action="http://www.javatpoint.com/javascriptpages/valid.jsp" onsubmit="return matchpass()">

Password:<input type="password" name="password" /><br/>

Re-enter Password:<input type="password" name="password2"/><br/>

<input type="submit">

</form>

</body>

</html>

JavaScript Number Validation

Let's validate the textfield for numeric value only. Here, we are using isNaN() function.

<!DOCTYPE html>

<html>

<head>

<script>

function validate(){

var num=document.myform.num.value;

if (isNaN(num)){

document.getElementById("numloc").innerHTML="Enter Numeric value only";

return false;

}else{

return true;

}

}

</script>

</head>

<body>

<form name="myform" action="http://www.javatpoint.com/javascriptpages/valid.jsp" onsubmit="return validate()" >

Number: <input type="text" name="num"><span id="numloc"></span><br/>

<input type="submit" value="submit">

</form>

</body>

</html>Bottom of Form

JavaScript email validation

We can validate the email by the help of JavaScript.

There are many criteria that need to be follow to validate the email id such as:

* email id must contain the @ and . character
* There must be at least one character before and after the @.
* There must be at least two characters after . (dot).

Let's see the simple example to validate the email field.

<html>

<body>

<script>

function validateemail()

{

var x=document.myform.email.value;

var atposition=x.indexOf("@");

var dotposition=x.lastIndexOf(".");

if (atposition<1 || dotposition<atposition+2 || dotposition+2>=x.length){

alert("Please enter a valid e-mail address \n atpostion:"+atposition+"\n dotposition:"+dotposition);

return false;

}

}

</script>

<body>

<form name="myform" method="post" action="http://www.javatpoint.com/javascriptpages/valid.jsp" onsubmit="return validateemail();">

Email: <input type="text" name="email"><br/>

<input type="submit" value="register">

</form>

</body>

</html>