

JavaScript: Validation des formulaires

Qu'est-ce que la validation d'un formulaire?

La validation de formulaire consiste à vérifier que les données fournies par l'utilisateur à l'aide d'un formulaire satisfont aux critères définis pour la collecte de données auprès des utilisateurs. Par exemple, si vous utilisez un formulaire d'inscription et que vous souhaitez que votre utilisateur fournisse son nom, son adresse e-mail et son adresse, vous devez utiliser un programme (en JavaScript ou dans toute autre langue) pour vérifier si l'utilisateur, fournisse une adresse e-mail valide et une adresse correcte.

Accès aux données d'un formulaire

Si un document HTML contient plusieurs formulaires, ils peuvent être accédés par `document.form_name` où `form_name` est la valeur de l'attribut `name` du formulaire ou par `document.forms [i]` où `i` est 0, 1,2, 3 et `document.forms [0]` fait référence au 1^{er} formulaire, `document.forms [1]` fait référence au 2^{ème} et ainsi de suite.

Elements of a form can be accessed by `document.form_name.form_element` where `form_name` is the value of the name attribute of the form element, `form_element` is the value of the name attribute of the form's element.

Les éléments d'un formulaire peuvent être accédés par **`document.form_name.form_element`** où **`form_name`** est la valeur de l'attribut `name` du formulaire et **`form_element`** est la valeur de l'attribut `name` de l'élément du formulaire.

Il est aussi possible d'utiliser le DOM pour accéder à un formulaire et à ses éléments.

JavaScript: vérifier qu'un champ text d'un formulaire est non vide

Souvent, des situations se présentent lorsqu'un utilisateur doit remplir un ou plusieurs champs dans un formulaire HTML avant de le soumettre. Vous pouvez écrire un script de validation de formulaire JavaScript pour vérifier si les champs obligatoires dans le formulaire HTML sont vides ou non.

La fonction suivante peut être utilisée pour vérifier si l'utilisateur a entré quelque chose dans un champ donné. Les champs vides indiquent deux types de valeurs. Une chaîne de longueur nulle ou une valeur NULL.

Javascript function to check whether a field is empty or not

```
// If the length of the element's string is 0 then display helper
message
function required(inputtx)
{
    if (inputtx.value.length == 0)
```

```

    {
        alert("message");
        return false;
    }
    return true;
}

```

Au début, la fonction `required ()` acceptera la valeur d'entrée HTML via le paramètre `inputx` en tant qu'objet. Après elle teste la longueur de chaîne, en utilisant la propriété `length` de l'objet paramètre. Si la longueur de `value.inputx` est 0 alors elle renvoie `false` sinon `true`.

HTML Code

```

<!doctype html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>JavaScript Validation de formulaire- controller qu'un champ
est nonvide</title>
<link rel='stylesheet' href='form-style.css' type='text/css' />
</head>
<body>
<div class="mail">
<h2>Saisir votre nom et envoyer</h2>
<form name="form1" action="#" onsubmit="required()">
<ul>
<li><input type='text' name ='text1'/></li>
<li class="rq">*Champ obligatoire</li>
<li><input type="submit" name="submit" value="Submit" /></li>
</ul>
</form>
</div>
<script src="non-empty.js"></script>
</body>
</html>

```

JavaScript Code

```

function required()
{
var empt = document.forms["form1"]["text1"].value;
if (empt == "")
{
alert("Prière de saisir une valeur");
return false;
}
else
{
alert('Valeur acceptée : vous pouvez faire un autre essai');
return true;
}
}

```

CSS Code

```
li {list-style-type: none;
font-size: 16pt;
}
.mail {
margin: auto;
padding-top: 10px;
padding-bottom: 10px;
width: 400px;
background : #D8F1F8;
border: 1px solid silver;
}
.mail h2 {
margin-left: 38px;
}
input {
font-size: 20pt;
}
input:focus, textarea:focus{
background-color: lightyellow;
}
input submit {
font-size: 12pt;
}
.rq {
color: #FF0000;
font-size: 10pt;
}
```

Une autre fonction pour controler qu'un champ est non vide

```
function Emptyvalidation(inputtxt)
{
if (inputtxt.value.length == 0)
{
document.inputtxt.style.background = 'Yellow';
}
else
{
document.inputtxt.style.background = 'White';
}
return error;
}
```

Dans la fonction ci-dessus, la couleur de fond du champ de saisie devient jaune si l'utilisateur laisse le champ vide, sinon la couleur de fond reste blanche.

JavaScript: Contrôler qu'on a saisi que des lettres

Parfois, des situations se présentent où l'utilisateur doit remplir un ou plusieurs champs alphabétiques (A-Z ou a-z) dans un formulaire HTML. Vous pouvez écrire un script de validation de formulaire JavaScript pour vérifier si les champs obligatoires dans le formulaire HTML contiennent uniquement des lettres.

Javascript function to check for all letters in a field

```
function allLetter(inputtxt)
{
  var letters = /^[A-Za-z]+$/;
  if(inputtxt.value.match(letters))
  {
    return true;
  }
  else
  {
    alert("message");
    return false;
  }
}
```

Pour obtenir une chaîne qui ne contient que des lettres (majuscules ou minuscules), nous utilisons une expression régulière (`/^[A-Za-z]+$/`) qui ne permet que des lettres. Ensuite, la méthode `match()` de l'objet chaîne est utilisée pour faire correspondre cette expression régulière avec la valeur d'entrée.

HTML Code

```
<!DOCTYPE html>
<html lang="fr">
<head>
<meta charset="utf-8">
<meta http-equiv="content-type" content="text/html; charset=utf-8"
/>
<title>JavaScript : Validation de formulaire- controler que des
lettres</title>
<link rel='stylesheet' href='form-style.css' type='text/css' />
</head>
<body onload='document.form1.text1.focus() '>
<div class="mail">
<h2>Entrer votre nom et soumettre</h2>
<form name="form1" action="#">
<ul>
<li>Code:</li>
<li><input type='text' name='text1' /></li>
```

```

<li class="rq">*Entrer uniquement des lettres.</li>
<li>&nbsp;</li>
<li><input type="submit" name="submit" value="Submit"
onclick="allLetter(document.form1.text1)" /></li>
<li>&nbsp;</li>
</ul>
</form>
</div>
<script src="all-letter.js"> </script>
</body>
</html>

```

JavaScript Code

```

function allLetter(inputtxt)
{
    var letters = /^[A-Za-z]+$/;
    if(inputtxt.value.match(letters))
    {
        alert('Votre nom a été accepté : Essayer un autre');
        return true;
    }
    else
    {
        alert('Prière de saisir uniquement des lettres');
        return false;
    }
}

```

JavaScript: Contrôler qu'il y a que des nombres

Parfois, des situations se présentent où l'utilisateur ne doit saisir que des chiffres (0-9) (numéro de téléphone, un code postal ou un numéro de carte de crédit) dans un champ d'un formulaire HTML. Vous pouvez écrire des scripts JavaScript pour vérifier l'usage :

- Seuls des entiers sans signe ont été saisi un dans un champ.
- Un entier avec un signe facultatif principal plus (+) ou moins (-).

Contrôler que seul un entier sans signe a été saisi

Pour vérifier qu'une chaîne ne contient que des chiffres (0-9) nous utilisons une expression régulière (/^[0-9]+\$/). Ensuite, la méthode match () de l'objet chaîne est utilisée pour faire correspondre cette expression régulière avec la valeur d'entrée.

HTML Code

```

<!DOCTYPE html>
<html lang="fr">

```

```

<head>
<meta charset="utf-8">
<title>JavaScript form validation - checking all numbers</title>
<link rel='stylesheet' href='form-style.css' type='text/css' />
</head>
<body onload='document.form1.text1.focus() '>
<div class="mail">
<h2>Input your registration number and submit</h2>
<form name="form1" action="#">
<ul>
<li><input type='text' name='text1' /></li>
<li class="rq">*Enter numbers only.</li>
<li>&nbsp;</li>
<li><input type="submit" name="submit" value="Submit"
onclick="allnumeric(document.form1.text1)" /></li>
<li>&nbsp;</li>
</ul>
</form>
</div>
<script src="all-numbers.js"></script>
</body>
</html>

```

JavaScript Code

```

function allnumeric(inputtxt)
{
    var numbers = /^[0-9]+$/;
    if(inputtxt.value.match(numbers))
    {
        alert('Votre enregistrement a été accepté....');
        document.form1.text1.focus();
        return true;
    }
    else
    {
        alert('Prière de saisir des chiffres uniquement');
        document.form1.text1.focus();
        return false;
    }
}

```

Vérifier qu'un entier avec un signe + ou - a été saisi

Pour vérifier qu'une chaîne ne contient que des chiffres (0-9) avec un signe facultatif + ou - (par exemple +12, -45, 12), nous utilisons l'expression régulière (/ ^ [- +]? [0-9] + \$ /). Ensuite, la méthode match () de l'objet chaîne est utilisée pour faire correspondre cette expression régulière avec la valeur d'entrée.

HTML Code

```

<!DOCTYPE html>
<html lang="fr">
<head>
<meta charset="utf-8">
<title>JavaScript form validation - An integer with an optional
leading plus(+) or minus(-) sign</title>
<link rel='stylesheet' href='form-style.css' type='text/css' />
</head>
<body onload='document.form1.text1.focus()>
<div class="mail">
<h2>Input your registration number and submit</h2>
<form name="form1" action="#">
<ul>
<li><input type='text' name='text1' /></li>
<li class="rq">*Enter numbers only.</li>
<li>&nbsp;</li>
<li><input type="submit" name="submit" value="Submit"
onclick="allnumericplusminus(document.form1.text1)" /></li>
<li>&nbsp;</li>
</ul>
</form>
</div>
<script src="all-numbers-with-plus-minus.js"></script>
</body>
</html>

```

JavaScript Code

```

function allnumericplusminus(inputtxt)
{
    var numbers = /^[+-]?[0-9]+$/;
    if(inputtxt.value.match(numbers))
    {
        alert('Correct...Try another');
        document.form1.text1.focus();
        return true;
    }
    else
    {
        alert('Please input correct format');
        document.form1.text1.focus();
        return false;
    }
}

```

JavaScript: Contrôler la saisie d'un nombre flottant

Un nombre flottant a les parties suivantes.

- Un nombre décimal entier.
- Un point décimal ('.').
- Une fraction.

- Un exposant (facultatif).

La partie exponentielle est un "e" ou un "E" suivi d'un entier, qui peut être signé (précédé de "+" ou "-").

Exemples de nombres flottant:

- 7.2935
- -12.72
- 1/2
- 12.4e3 [= 12.4 x 10³]
- 4E-3 [= 4 x 10⁻³]

Parfois, il est nécessaire d'entrer un nombre avec une partie décimale en particulier pour la quantité, la longueur, la hauteur, les montants, etc. Nous pouvons utiliser un nombre à virgule flottante de diverses façons, voici quelques exemples.

- Aucun exposant, nombre entier obligatoire et fraction et signe facultatif.
- Aucun exposant, signe obligatoire (+ -), nombre entier et fraction.
- Signe obligatoire (+ -), exposant, entier, fraction.

Pour vérifier le format, nous avons utilisé les différentes expressions régulières.

Fonction Javascript pour vérifier si une entrée de champ contient un nombre sans exposant, un entier obligatoire et une fraction et un signe facultatif

Pour vérifier qu'une chaîne contient un nombre sans exposant, entier obligatoire et fraction et signe facultatif, nous utilisons l'expression régulière / ^[-+]?[0-9]+\.[0-9]+\$/ . Ensuite, la méthode match () d'un objet chaîne est utilisée pour faire correspondre cette expression régulière avec la valeur d'entrée.

HTML Code

```
<!DOCTYPE html>
<html lang="fr">
<head>
<meta charset="utf-8">
<title>JavaScript: verifier un nombre flottant</title>
<link rel='stylesheet' href='form-style.css' type='text/css' />
</head>
<body onload='document.form1.text1.focus() '>
<div class="mail">
<h2>Check whether the field input contains a number with no
exponent, mandatory integer and fraction and optional sign</h2>
<form name="form1" action="#">
<ul>
<li><input type='text' name='text1' /></li><li>&nbsp;</li>
<li><input type="submit" name="submit" value="Submit"
onclick="CheckDecimal(document.form1.text1)" /></li>
<li>&nbsp;</li>
</ul>
```



```
</form>
</div>
<script src="check-decimal.js"></script>
</body>
</html>
```

JavaScript Code

```
function CheckDecimal(inputtxt)
{
var decimal= /^[+-]?[0-9]+\.[0-9]+$/;
if(inputtxt.value.match(decimal))
{
alert('Correct, try another...')
return true;
}
else
{
alert('Wrong...!')
return false;
}
}
```

Fonction Javascript pour vérifier si une entrée de champ contient un nombre sans exposant, un signe obligatoire (+ -), un entier et une fraction

Pour vérifier qu'une chaîne contient un nombre sans exposant, entier obligatoire, fraction, et signe (+ -) nous utilisons une expression régulière `/[+-]?[0-9]+\.[0-9]+/` qui permet le dit format. Ensuite, la méthode `match ()` d'un objet chaîne est utilisée pour faire correspondre cette expression régulière avec la valeur d'entrée.

HTML Code

```
<!DOCTYPE html>
<html lang="fr">
<head>
<meta charset="utf-8">
<title>JavaScript form validation - check a Floating Number starting with
sign</title>
<link rel='stylesheet' href='form-style.css' type='text/css' />
</head>
<body onload='document.form1.text1.focus() '>
<div class="mail">
<h2>Check whether a field input contains a number with no exponent, mandatory
sign (+-), integer, and fraction</h2>
<form name="form1" action="#">
<ul>
<li><input type='text' name='text1' /></li><li>&nbsp;</li>
<li><input type="submit" name="submit" value="Submit"
onclick="CheckDecimal(document.form1.text1)" /></li>
<li>&nbsp;</li>
</ul>
```

```
</form>
</div>
<script src="check-decimal-with-sign.js"></script>
</body>
</html>
```

JavaScript Code

```
function CheckDecimal(inputtxt)
{
var decimal= /^[+-][0-9]+\.[0-9]+$/;
if(inputtxt.value.match(decimal))
{
alert('Correct, try another...')
return true;
}
else
{
alert('Wrong...!')
return false;
}
}
```

Fonction Javascript pour vérifier si une entrée de champ contient un nombre commençant par le signe obligatoire (+ -), l'exposant, l'entier, et la fraction

Pour vérifier qu'une chaîne contient un nombre avec un signe obligatoire (+ -), exposant entier, fraction, un signe, nous utilisons l'expression régulière `/^[+-][0-9]+\.[0-9]+[eE][- +]? [0-9]+$/` qui permet le dit format. Ensuite, la méthode `match()` d'un objet chaîne est utilisée pour faire correspondre cette expression régulière avec la valeur d'entrée.

HTML Code

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>JavaScript form validation - check a Floating Number with exponent and
starting with sign </title>
<link rel='stylesheet' href='form-style.css' type='text/css' />
</head>
<body onload='document.form1.text1.focus()>
<div class="mail"><h2>Check whether a field input contains a number starting
with mandatory sign (+-), exponent, integer, fraction</h2>
<form name="form1" action="#">
<ul>
<li><input type='text' name='text1' /></li><li>&nbsp;</li>
<li><input type="submit" name="submit" value="Submit"
onclick="CheckDecimal(document.form1.text1)" /></li>
<li>&nbsp;</li>
</ul>
</form>
</div>
<script src="check-decimal-with-exponent-sign.js"></script>
</body>
</html>
```

JavaScript Code

```
function CheckDecimal(inputtxt)
var decimal= /^[+-][0-9]+\.[0-9]+[eE][+-]?[0-9]+$/;
if(inputtxt.value.match(decimal))
{
alert('Correct, try another...')
return true;
}
else
{
alert('Wrong...!')
return false;
}
}
```

JavaScript: Vérifier qu'une entrée ne contient que des chiffres et des lettres

Fonction Javascript

```
// Function to check letters and numbers
function alphanumeric(inputtxt)
{
var letterNumber = /^[0-9a-zA-Z]+$/;
if((inputtxt.value.match(letterNumber)))
{
return true;
}
else
{
alert("message");
return false;
}
}
```

HTML Code

```
<!DOCTYPE html>
<html lang="en">
<head>
<meta charset="utf-8">
<title>JavaScript form validation - checking all letters and numbers</title>
<link rel='stylesheet' href='form-style.css' type='text/css' />
</head>
<body onload='document.form1.text1.focus()>
<div class="mail">
<h2>Enter your Registration Number and Submit</h2>
<form name="form1" action="#">
<ul>
<li><input type='text' name='text1' /></li>
<li class="rq">*Enter numbers and alphabets only.</li>
<li>&nbsp;</li>
```

```
<li><input type="submit" name="submit" value="Submit"
onclick="alphanumeric(document.form1.text1)" /></li>
<li>&nbsp;</li>
</ul>
</form>
</div>
<script src="check-letters-numbers.js"></script>
</body>
</html>
```

Javascript code

```
function alphanumeric(inputtxt)
{
var letters = /^[0-9a-zA-Z]+$/;
if(inputtxt.value.match(letters))
{
alert('Your registration number have accepted : you can try
another');
document.form1.text1.focus();
return true;
}
else
{
alert('Please input alphanumeric characters only');
return false;
}
}
```

JavaScript: Vérification de la longueur d'une entrée

Fonction Javascript pour restreindre la longueur d'une entrée

```
function lengthRange(inputtxt, minlength, maxlength)
{
var userInput = inputtxt.value;
if(userInput.length >= minlength && userInput.length <=
maxlength)
{
return true;
}
else
{
alert("Please input between " +minlength+ " and " +maxlength+
" characters");
return false;
}
}
```

Voici un exemple qui utilise la fonction ci-dessus pour un champ qui nécessite 6 à 8 caractères pour valider un code utilisateur.

HTML Code

```
<!DOCTYPE html>
```

```

<html lang="fr">
<head>
<meta charset="utf-8">
<title>JavaScript form validation - checking non empty</title>
<link rel='stylesheet' href='form-style.css' type='text/css' />
</head>
<body onload='document.form1.text1.focus() '>
<div class="mail">
<p>Enter Userid [between 6 to 8 characters] and Submit</p>
<form name="form1" action="#">
<ul>
<li>Username:<input type='text' name='text1' /></li>
<li>&nbsp;</li>
<li class="submit"it"><input type="submit" name="submit"
value="Submit"
onclick="stringlength(document.form1.text1,6,8)"/></li>
<li>&nbsp;</li>
</form>
</ul>
</div>
<script src="string-lenght.js"></script>
</body>
</html>

```

Javascript code

```

function stringlength(inputtxt, minlength, maxlength)
{
var field = inputtxt.value;
var mlen = minlength;
var mxlen = maxlength;

if(field.length<mlen || field.length> mxlen)
{
alert("Please input the userid between " +mlen+ " and " +mxlen+ "
characters");
return false;
}
else
{
alert('Your userid have accepted. ');
return true;
}
}

```

JavaScript: Validation d'un email

La validation d'un email est un point très important lors de la validation d'un formulaire HTML.

Un e-mail est une chaîne séparée en deux parties par le symbole @. un "personal_info" et un domaine, c'est-à-dire personal_info @ domain. La longueur de la partie personal_info peut comporter jusqu'à 64 caractères et le nom de domaine peut comporter jusqu'à 253 caractères.

La partie personal_info contient les caractères ASCII suivants.

- Lettres anglaises majuscules (A-Z) et minuscules (a-z).
- Chiffres (0-9).
- Caractères spéciaux ! # \$ % & ' * + - / = ? ^ _ ` { | } ~
- Le point(.) à condition que ce ne soit pas le premier ou le dernier caractère et qu'il ne vienne pas l'un après l'autre.

La partie du nom de domaine [par exemple com, org, net, in, us, info] contient des lettres, des chiffres, des tirets et des points.

Exemple d'email valide

- mysite@mydom.com
- my.ownsite@dom.org
- mysite@you.me.net

Exemple d'emails invalides

- mysite.dom.com [@ absent]
- mysite@.com.my [domaine commence par un point « . »]
- @you.me.net [personal-info absente @]
- mysite123@gmail.b [".b" n'est pas un domaine valide]
- mysite@.org.org [tdomaine ne doit pas commencer par un point]
- .mysite@mysite.org [un email ne doit pas commencer par un "."]
- mysite..1234@yahoo.com [deux points se suivent]

JavaScript code to validate an email id

```
function ValidateEmail(mail)
{
  if (/^\w+([\.-]?\w+)*@\w+([\.-]
]? \w+)*(\.\w{2,3})+$/ .test(myForm.emailAddr.value))
  {
    return (true)
  }
  alert("You have entered an invalid email address!")
  return (false)
}
```

Pour vérifier un email nous utilisons l'expression régulière :

`/^\w+([\.-]?\w+)*@\w+([\.-]?\w+)*(\.\w{2,3})+$/`

Explication des caractères utilisés dans l'expression régulière

`/^\w+([\.-]?\w+)*@\w+([\.-]?\w+)*(\.\w{2,3})+$/`

Character	Description
<code>/.. /</code>	All regular expressions start and end with forward slashes.
<code>^</code>	Matches the beginning of the string or line.
<code>\w+</code>	Matches one or more word characters including the underscore. Equivalent to <code>[A-Za-z0-9_]</code> .
<code>[\.-]</code>	<code>\</code> Indicates that the next character is special and not to be interpreted literally. <code>.-</code> matches character <code>.</code> or <code>-</code> .
<code>?</code>	Matches the previous character 0 or 1 time. Here previous character is <code>[\.-]</code> .
<code>\w+</code>	Matches 1 or more word characters including the underscore. Equivalent to <code>[A-Za-z0-9_]</code> .
<code>*</code>	Matches the previous character 0 or more times.
<code>([\.-]?\w+)*</code>	Matches 0 or more occurrences of <code>[\.-]?\w+</code> .
<code>\w+([\.-]?\w+)*</code>	The sub-expression <code>\w+([\.-]?\w+)*</code> is used to match the username in the email. It begins with at least one or more word characters including the underscore, equivalent to <code>[A-Za-z0-9_]</code> . <code>.</code> or <code>-</code> and <code>.</code> or <code>-</code> must follow by a word character (<code>A-Za-z0-9_</code>).
<code>@</code>	It matches only <code>@</code> character.
<code>\w+([\.-]?\w+)*</code>	It matches the domain name with the same pattern of user name described above..
<code>\.\w{2,3}</code>	It matches a <code>.</code> followed by two or three word characters, e.g., <code>.edu</code> , <code>.org</code> , <code>.com</code> , <code>.uk</code> , <code>.us</code> , <code>.co</code> etc.
<code>+</code>	The <code>+</code> sign specifies that the above sub-expression shall occur one or more times, e.g., <code>.com</code> , <code>.co.us</code> , <code>.edu.uk</code> etc.
<code>\$</code>	Matches the end of the string or line.

Note: Si vous voulez travailler sur un domaine à 4 chiffres, par exemple, `.info`, vous devez changer `w{2,3}` avec `w{2,4}`.

HTML Code
<pre> <!DOCTYPE html> <html lang="fr"> <head> <meta charset="utf-8"> <title>JavaScript form validation - checking email</title> <link rel='stylesheet' href='form-style.css' type='text/css' /> </pre>

```

</head>
<body onload='document.form1.text1.focus() '>
<div class="mail">
<h2>Input an email and Submit</h2>
<form name="form1" action="#">
<ul>
<li><input type='text' name='text1' /></li>
<li>&nbsp;</li>
<li class="submit"><input type="submit" name="submit"
value="Submit" onclick="ValidateEmail(document.form1.text1)" /></li>
<li>&nbsp;</li>
</ul>
</form>
</div>
<script src="email-validation.js"></script>
</body>
</html>

```

JavaScript Code

```

function ValidateEmail(inputText)
{
var mailformat = /^^\w+([\.-]?\w+)*@\w+([\.-]?\w+)*(\.\w{2,3})+$/;
if(inputText.value.match(mailformat))
{
document.form1.text1.focus();
return true;
}
else
{
alert("You have entered an invalid email address!");
document.form1.text1.focus();
return false;
}
}

```

JavaScript: Validation d'une Date

Dans le code qui suit nous validons les dates qui ont l'une des formats suivants:

1. dd/mm/yyyy or dd-mm-yyyy format.
2. mm/dd/yyyy or mm-dd-yyyy format.

Validation du format dd/mm/yyyy ou dd-mm-yyyy

HTML Code

```

<!DOCTYPE html>
<html lang="fr">
<head>
<meta charset="utf-8">
<title>JavaScript form validation - checking date</title>
<link rel='stylesheet' href='form-style.css' type='text/css' />
</head><body onload='document.form1.text1.focus() '>

```



```

<div class="mail">
<h2>Input a valid date [dd/mm/yyyy or dd-mm-yyyy format]</h2>
<form name="form1" action="#">
<ul>
<li><input type='text' name='text1' /></li>
<li>&nbsp;</li>
<li class="submit"><input type="submit" name="submit"
value="Submit" onclick="validateddate(document.form1.text1)" /></li>
<li>&nbsp;</li>
</ul>
</form>
</div>
<script src="ddmmyyyy-validation.js"></script>
</body>
</html>

```

JavaScript Code

```

function validateddate(inputText)
{
    var dateformat = /^(0?[1-9]|[12][0-9]|3[01])[\/\-](0?[1-9]|1[012])[\/\-
]\d{4}$/;
    // Match the date format through regular expression
    if(inputText.value.match(dateformat))
    {
        document.form1.text1.focus();
        //Test which separator is used '/' or '-'
        var opera1 = inputText.value.split('/');
        var opera2 = inputText.value.split('-');
        lopera1 = opera1.length;
        lopera2 = opera2.length;
        // Extract the string into month, date and year
        if (lopera1>1)
        {
            var pdate = inputText.value.split('/');
        }
        else if (lopera2>1)
        {
            var pdate = inputText.value.split('-');
        }
        var dd = parseInt(pdate[0]);
        var mm = parseInt(pdate[1]);
        var yy = parseInt(pdate[2]);
        // Create list of days of a month [assume there is no leap year by default]
        var ListofDays = [31,28,31,30,31,30,31,31,30,31,30,31];
        if (mm==1 || mm>2)
        {
            if (dd>ListofDays[mm-1])
            {
                alert('Invalid date format!');
                return false;
            }
        }
        if (mm==2)
        {
            var lyyear = false;
            if ( (!yy % 4) && yy % 100) || !(yy % 400))
            {
                lyyear = true;
            }
        }
    }
}

```

```

if ((lyear==false) && (dd>=29))
{
alert('Invalid date format!');
return false;
}
if ((lyear==true) && (dd>29))
{
alert('Invalid date format!');
return false;
}
}
}
else
{
alert("Invalid date format!");
document.form1.text1.focus();
return false;
}
}
}

```

Validation du format mm/dd/yyyy ou mm-dd-yyyy

HTML Code

```

<!DOCTYPE html>
<html lang="en"><head>
<meta charset="utf-8">
<title>JavaScript form validation - checking date [mm/dd/yyyy or
mm-dd-yyyy format]</title>
<link rel='stylesheet' href='form-style.css' type='text/css' />
</head>
<body onload='document.form1.text1.focus()>
<div class="mail">
<h2>Input a valid date [mm/dd/yyyy or mm-dd-yyyy format]</h2>
<form name="form1" action="#">
<ul>
<li><input type='text' name='text1' /></li>
<li>&nbsp;</li>
<li class="submit"><input type="submit" name="submit"
value="Submit" onclick="validatedate(document.form1.text1)" /></li>
<li>&nbsp;</li>
</ul>
</form>
</div>
<script src="mmdyyyy-validation.js"></script>
</body>
</html>

```

JavaScript Code

```

function validatedate(inputText)
{
var dateformat = /^(0?[1-9]|1[012])[\-\/\-](0?[1-9]|12)[0-9]|3[01])[\-\/\-]
\d{4}$/;
// Match the date format through regular expression
if(inputText.value.match(dateformat))
{
document.form1.text1.focus();
}
}

```

```

//Test which separator is used '/' or '-'
var opera1 = inputText.value.split('/');
var opera2 = inputText.value.split('-');
lopera1 = opera1.length;
lopera2 = opera2.length;
// Extract the string into month, date and year
if (lopera1>1)
{
var pdate = inputText.value.split('/');
}
else if (lopera2>1)
{
var pdate = inputText.value.split('-');
}
var mm = parseInt(pdate[0]);
var dd = parseInt(pdate[1]);
var yy = parseInt(pdate[2]);
// Create list of days of a month [assume there is no leap year by default]
var ListofDays = [31,28,31,30,31,30,31,31,30,31,30,31];
if (mm==1 || mm>2)
{
if (dd>ListofDays[mm-1])
{
alert('Invalid date format!');
return false;
}
}
if (mm==2)
{
var lyear = false;
if ( (!yy % 4) && yy % 100) || !(yy % 400))
{
lyear = true;
}
if ((lyear==false) && (dd>=29))
{
alert('Invalid date format!');
return false;
}
if ((lyear==true) && (dd>29))
{
alert('Invalid date format!');
return false;
}
}
}
else
{
alert("Invalid date format!");
document.form1.text1.focus();
return false;
}
}
}

```

JavaScript: validation d'un numéro de téléphone

Le code ci-dessous valide un numéro de téléphone qui contient 10 chiffres

Code JavaScript

```

function phonenumber(inputtxt)
{
var phoneno = /^d{10}$/;

```

```

if((inputtxt.value.match(phoneno))
{
return true;
}
else
{
alert("message");
return false;
}
}

```

Pour valider un numéro qui a un des formats suivants:

XXX-XXX-XXXX

XXX.XXX.XXXX

XXX XXX XXXX

utiliser le code ci-dessous.

```

function phonenumber(inputtxt)
{
var phoneno = /^\(?(?([0-9]{3})\)?[-. ]?(?([0-9]{3})[-. ]?(?([0-9]{4})$)/;
if((inputtxt.value.match(phoneno))
{
return true;
}
else
{
alert("message");
return false;
}
}
}

```

Si vous souhaitez utiliser un signe + avant le numéro de la manière suivante

+XX-XXXX-XXXX

+XX.XXXX.XXXX

+XX XXXX XXXX

utiliser le code.

```

function phonenumber(inputtxt)
{
var phoneno = /^\+?(?([0-9]{2})\)?[-. ]?(?([0-9]{4})[-. ]?(?([0-9]{4})$)/;
if((inputtxt.value.match(phoneno))
{
return true;
}
else
{
alert("message");
return false;
}
}
}

```

```
}
```

Validation d'un numéro de telephone international avec le code du pays

Le code ci-dessous montre comment valider un numéro de téléphone avec le code du pays, qui a le format +24-0455-9034, +21.3789.4512 ou +23 1256 4587 .

HTML Code

```
<!DOCTYPE html><html lang="en"><head>
<meta charset="utf-8">
<title>JavaScript form validation - checking non-empty</title>
<link rel='stylesheet' href='form-style.css' type='text/css' />
</head>

<body onload='document.form1.text1.focus() '>
<div class="mail">
<h2>Input an Phone No.[+xx-xxxx-xxxx, +xx.xxxx.xxxx, +xx xxxx xxxx]
and Submit</h2>
<form name="form1" action="#">
<ul>
<li><input type='text' name='text1' /></li>
<li>&nbsp;</li>
<li class="submit"><input type="submit" name="submit"
value="Submit" onclick="phonenumber(document.form1.text1)" /></li>
<li>&nbsp;</li>
</ul>
</form>
</div>
<script src="phonenumber-international-format.js"></script>
</body>
</html>
```

JavaScript Code

```
function phonenumber(inputtxt)
{
  var phoneno = /^\(?(?([0-9]{3})\)?[-. ]?(?([0-9]{3})[-. ]?(?([0-9]{4})$)/;
  if(inputtxt.value.match(phoneno))
  {
    return true;
  }
  else
  {
    alert("Not a valid Phone Number");
    return false;
  }
}
```