**Plugin Jquery validate :**

Librairie qui nous permet de faire la validation côté client en JavaScript. Attention cette méthode ne nous permet pas d’omettre la validation côté serveur.

La librairie a la particularité de reprendre un sujet auquel nous devons régulièrement faire face soit la validation de nos formulaires. Pour sauver du temps et de l’énergie Au lieu de refaire cette validation, nous pouvons profiter d’une librairie qui a prévu plusieurs types de validation.

Liens intéressants :

<https://www.pierrefay.fr/blog/jquery-validate-formulaire-validation-tutoriel.html>

<http://fr.openclassrooms.com/informatique/cours/simplifiez-vos-developpements-javascript-avec-jquery/trouver-et-utiliser-un-plugin>

Pour utiliser la librairie, il suffit de se rendre sur <http://jqueryvalidation.org/> , de downloader la librairie (dernière version) et de la décompresser.

Vous avez des démos qui vous guident dans l’utilisation de la librairie et de ses possibilités.

Je vous propose d’utiliser la librairie de 2 façons :

1. Directement dans le code HTML du formulaire
2. Par des déclarations JavaScript dans la page (permettant d’ajouter des REGEXP)

Il faudra donc inclure les librairies nécessaires jquery , jquery validate et les messages modifiables:

<**script** type="text/javascript" src="lib/jquery-1.11.1.js"></**script**>

<**script** type="text/javascript" src="dist/jquery.validate.min.js"></**script**>

<**script** type="text/javascript" src="dist/localization/messages\_fr.js"></**script**>

1. **Utilisation directement dans le code HTML du formulaire :**

On se déclare un formulaire, on y ajoute par exemple un textbox. Nous ajouterons la validation pour que le champ soit obligatoire et qu’il ait au moins 4 caractères :

<**form** class="mon\_formu" id="mon\_formu" method="Post" action="test.php">

<p>

<label for="nickname">surnom (au moins 4 caractères)</label>

<input id="nickname" name="nickname" **minlength**="4" type="text" **required**/>

</p>

<p><input class="submit" type="submit" value="Se connecter"/></p>

</**form**>

Pour compléter, il faut spécifier l’évènement **validate** sur le formulaire :

$("#**mon**\_**formu**").**validate**();

Il vous est possible de voir en plus des validations possibles, les messages qui seront affichés dans message\_fr.js :

$.extend($.**validator**.**messages**, {

required: "Ce champ est obligatoire.",

remote: "Veuillez corriger ce champ.",

email: "Veuillez fournir une adresse électronique valide.",

url: "Veuillez fournir une adresse URL valide.",

date: "Veuillez fournir une date valide.",

**…**

1. **Par des déclarations JavaScript dans la page (permettant d’ajouter des REGEXP)**

Cependant, quelques fois les validations ne répondent pas exactement à nos besoins, il nous est alors possible d’appliquer des expressions régulières.

Nous allons nous ajouter un champ dans le formulaire, sans spécifier de validation :

<p>

<label for="taille"> taille (requis)</label>

<input id=" taille " name=" taille " type="text" />

</p>

Nous pourrons ajouter des règles de validation et des messages personnalisés dans notre événement. Le champ taille sera **required** et devra être entre 0 et 200 inclus. Les messages qui s’afficheront seront personnalisés.

$("#mon\_formu").**validate**({

**rules**: {

nickname : {

required : true,

minlength : 4

},

taille : {

required : true,

min : 0,

max : 200,

},

},

**messages**: {

nickname : {

required : 'Le surnom est obligatoire',

minlength : 'le surnom doit avoir au moins 4 caractères'

}

}

});

Pour ajouter une REGEXP, il faut ajouter une méthode :

$.validator.**addMethod**("regexp\_monchamp",

function (value, element){

return this.optional(element) || **/^[a-zA-Z}$/.test(value)**;

}, 'Vous devez respecter le format demandé');

Et ajouter cette règle à notre champ :

monchamp :{

**required** :true,

**regexp\_monchamp**:true,

}

Avec jquery validate plusieurs validations sont possibles :

Required, Minlength, Maxlength, Rangelength, Min, Max, Range, Email, url, date, number, creditcard, equalto, digits,…

Toutes expressions régulières (REGEXP)

Voir documentation sur les méthodes de validation :

<http://jqueryvalidation.org/documentation/>

N.B.

Pour forcer la validation en sortant du champ on peut ajouter onfocusout avant les rules.

<script>

$("#mon\_formu").validate({

**onfocusout: function(maFonction) {**

**this.element(maFonction);**

},

rules:

…

On peut changer la couleur des messages avec errorClass

$("#mon\_formu").validate({

**errorClass**: "my-error-class",

rules:

…

puis on se définit un style :

<style>

.my-error-class {Color :red;}

</style>