

# Les Médecins Maîtres-Toile

[Accueil](#) ▶ [Boite à Outils](#) ▶ Réaliser un formulaire de contact (3/4)

**Publié le : 3 février 2004**

 [Imprimer cet article](#)

**Auteur :**  
**Philippe Auriol**



## Réaliser un formulaire de contact (3/4)

**html, css, javascript, php**

**Rien de plus désagréable que de recevoir un formulaire de la part de quelqu'un que l'on ne connaît pas et que l'on ne sait pas comment joindre. Si un expéditeur vous envoie par la poste un courrier sans adresse de retour vous n'y pouvez rien, heureusement sur Internet, vous pouvez quand même accompagner un peu plus votre visiteur pour l'aider à mieux faire les choses.**

Javascript est un langage qui s'exécute "côté client", c'est à dire sur le navigateur de votre visiteur. Il vient donc s'ajouter dans le code de la page html et imbrique ainsi des actions alors même que l'ordinateur n'est pas forcément connecté à l'internet.

Dans le cas d'un formulaire envoyé à des collègues et qui peut être rempli "hors ligne" avant d'être envoyé "en-ligne" le fait que ce langage soit interprété par le navigateur lui-même est un atout indéniable.

Il aurait été possible de vérifier les champs du formulaire "en ligne" avec le javascript aussi, ou pour les navigateurs ayant désactivé le javascript de le faire contrôler par du php (donc du côté du serveur Internet).

### Javascript et verification() sont dans un formulaire...

Pour que le script s'exécute quand le formulaire sera validé, il nous faut placer d'abord dans la balise `<form ...>` le texte suivant `onSubmit="return verification() ">` ce qui donnera :

```
<form name="formulaire" action="traitement.php" method="post" onSubmit="return verification() ">
```

Il nous faut maintenant nous intéresser aux champs eux-mêmes et surtout à leurs noms et types.

trois champs texte dont un pour l'email  
un champ radio  
un champ textarea

Un nom de champ de formulaire en javascript s'écrit ainsi :

`document.name_du_formulaire.nameduchamp` donc pour le nom de l'expéditeur (`name="nom"`) et dans le cas de notre formulaire dont le nom (`name`) est "formulaire" (

document.formulaire.nom

Si vous voulez en savoir plus sur le nommage en javascript, je vous invite à [consulter le site de référence sur le sujet](#). Et si vous souhaitez pousser plus loin votre apprentissage du javascript [le site de Bernard Martin Raibault](#) devrait vous ravir.

### **Et alors ? A quoi ça sert de nommer un morceau du formulaire ?**

Ne soyez pas impatients ! Pouvoir mettre un nom sur une chose, c'est déjà la connaître.

Alors, plaçons dans le head de notre page html le code suivant :

```
<script type="text/javascript" language="Javascript" >
<!--
function verification()
{
}
//-->
</script>
```

Cela dit à notre navigateur qu'il y a ici un script à lire, le langage dans lequel il est écrit et les commentaires "" indiquent au navigateur ne connaissant pas le javascript de l'ignorer et de ne pas afficher d'erreur.

Après les commentaires, voilà le nom de la fonction que nous appelons "verification" puis arrivent des accolades dans lesquelles nous allons placer ce que doit faire le script.

### **Et comment je sais que le monsieur il a mis son nom dans le formulaire ?**

Pas compliqué : on demande poliment à javascript de le faire à notre place et ça donne :

```
if(document.formulaire.nom.value == "") {
    alert("Veuillez entrer votre Nom svp");
    document.formulaire.nom.focus();
    return false;
}
else
return true;
```

Et voilà : le script regarde si le champ document.formulaire.nom est égal à rien ("") et si c'est le cas il affiche une alerte demandant de remplir ce champ et, comme c'est un champ texte, il renvoie le "focus" vers ce champ et empêche l'envoi du formulaire (return false).

Et le prénom ? pareil. Sauf que l'on met "prenom" au lieu de "nom"...

### **Ok mais euh...le mail ?**

Le mail c'est aussi un champ texte, donc on peut très bien se contenter de tester si le champ est vide ou pas.

On peut aussi ne pas s'en contenter : ok.

---

Alors...testons si il ya un @ dans ce champ :

```
else
if(document.formulaire.email.value == "") {
    alert("Veuillez entrer votre adresse electronique svp");
    document.formulaire.email.focus();
    return false;
}
else
if(document.formulaire.email.value.indexOf('@') == -1) {
    alert("Ce n'est pas une adresse electronique, verifiez la svp");
    document.formulaire.email.focus();
    return false;
}
```

Et là tout nom avec un @ sera validé. On peut aller bien plus loin en vérifiant des tas de choses dans l'email mais ça se complique notablement : je vous invite à étudier alors les "regexp" qui servent à cela et pas seulement en javascript.

Bon, voilà trois champs de faits.

**Et les cases à cocher ? Vous y avez pensé aux cases à cocher ?**

Effectivement, cela n'a pas grand chose à voir : comment les tester ces fichus cases à cocher ? Il y en a trois et elles ont le même nom en plus...normal : elles correspondent à UN choix parmi les trois.

Le problème serait d'ailleurs le même avec des champs select ou des cases à cocher (choix multiples).

En javascript on testera ça comme ça :

```
else
    sujetoption = -1
    for (i=0; i<formulaire.sujet.length; i++)
    {
        if(document.formulaire.sujet[i].checked){
            sujetoption = i
        }
    }
    if (sujetoption == -1) {
        alert("Veuillez choisir un sujet pour votre mail svp");
        return false;
    }
```

On donne une valeur (-1) à une variable sujetoption, on compte le nombre de champs sujet existant et on regarde si il y en a au moins un de coché, si c'est le cas on change la valeur de sujetoption pour i (le numéro de rang de la case cochée) après cela on teste combien vaut sujetoption : s'il vaut toujours "-1", c'est pas coché et donc on envoie un message d'erreur, si c'est coché il renvoie une valeur autre et on passe à l'argument suivant de notre fonction !

**Et voilà, la fonction est finie !**

---

Comment ça non ? Ah oui, le champ de texte. Bah, c'est facile : vous l'avez déjà fait.

ça donne donc

```
else
if(document.formulaire.texte.value == "") {
    alert("Veuillez entrer le texte de votre message svp");
    document.formulaire.texte.focus();
    return false;
}
```


Et voilà, cette fois c'est fini. Il ne nous reste plus qu'à faire le traitement du formulaire en php, [c'est la quatrième partie](#)

### Document joint à l'article



le formulaire complet - html+css+js = fini !



 [Imprimer cet article](#)

Copyright Médecins Maîtres-Toile francophones  
[Espace membres](#) - [Administration](#) - [Crédits](#)

---