

Les Médecins Maîtres-Toile

[Accueil](#) ▶ [Boîte à Outils](#) ▶ [Aide SPIP](#) ▶ Exécuter un Javascript dans une page SPIP

Publié le : 20 mai 2008
Publication antérieure :
11 décembre 2004

 **Imprimer cet article**

Auteur :
Alain Sichel †



Exécuter un Javascript dans une page SPIP

Cet article explique comment utiliser JavaScript dans une page SPIP. Il est complètement dépassé maintenant puisque depuis la version 1.9 de Spip il est possible d'utiliser Javascript. A n'utiliser donc que pour les versions antérieures... mais est-ce bien raisonnable d'utiliser une ancienne version de Spip ?

Sommaire

- [Pour quoi faire](#)
- [Est-ce bien raisonnable](#)
- [Modifier le squelette](#)
- [Contourner la sécurité](#)
- [Avertissements](#)
- [Et PHP](#)

Pour quoi faire ?

Cet article fait partie des essais que je mène actuellement dans le but de passer le site du [CYES](#) sous Spip : j'ai cherché une solution pour que les scripts JavaScript que j'utilise actuellement puissent continuer à fonctionner.

Est-ce bien raisonnable ?

Il y a dans Spip une sécurité qui fait que si on écrit `<script <? <?php` ou `<%` le caractère `<` sera remplacé par son équivalent HTML `<`; ce qui fait que le script ne peut pas fonctionner (par contre son contenu sera affiché comme du texte). Cette sécurité est importante pour contrôler le site : le

responsable du site doit savoir si un script s'exécute dans ses pages. Avec ce système tout script est a priori interdit.

La suite de cet article explique comment contourner cette sécurité, mais avant cela, on doit se demander si c'est bien nécessaire. Je ne suis qu'un débutant "en Spip" et j'ignore sûrement des solutions possibles par une bonne utilisation des filtres, des boucles ou des mots clés. On peut aussi trouver d'autres solutions qu'un script, ainsi par exemple, un rollover sur une image peut se faire en utilisant les CSS [1].

Modifier le squelette

Une solution simple est de noter les scripts dans le squelette :

► Pour tous les articles du site :

C'est une bonne solution par exemple pour un menu de navigation utilisé dans toutes les pages du site. Il faut pour cela modifier le squelette des articles, c'est-à-dire le fichier "*article.html*" [2]. Vous placez votre script entre `<head>` et `</head>`, soit le code complet du script, soit sous la forme d'un fichier externe :

```
<script language="JavaScript" type="text/javascript"
src="fichier_externe.js"></script>
```

Il faut ensuite placer dans le corps de la page le code qui lancera la fonction JavaScript,

- dans le body, par exemple : `<body onload="Chargement();">`
- ou à un autre endroit de la page *article.html* : `<script language="javascript">Chargement();</script>`

On peut aussi choisir de ne pas mettre le lancement de la fonction JavaScript dans "*article.html*" mais dans le champ **Texte** de certains articles [3], par exemple : `` [4] Pour que cela fonctionne, il faut mettre dans votre article ce code entre deux tags `<html>` et `</html>`.

► Pour un article particulier :

Les mêmes principes peuvent être utilisés pour un (ou plusieurs) article(s), il faudra alors créer un squelette spécifique à partir du fichier "*article.html*", par exemple *article=25.html* si on veut utiliser JavaScript uniquement pour l'article n° 25, et le modifier comme indiqué précédemment en plaçant le script entre `<head>` et `</head>`. Mais ce n'est pas fini, car on ne peut pas faire dans Spip un squelette spécifique pour un article comme on peut le faire pour une rubrique, pour que ça marche, il faut modifier le fichier "*article.php3*" [5] en remplaçant `$fond = "article";` par

```
if($id_article=="25") {$fond = "article=25";}
else {$fond = "article";}
```

Contourner la sécurité

S'il faut vraiment mettre un Javascript dans un article, il faut contourner la sécurité qui transforme `<script...>` en `<script...>`. Il faut pour cela modifier le fichier "*inc_texte.php3*" dans le dossier "*ecrire*" de votre site. Avant cela, je vous conseille de sauvegarder le fichier d'origine, par exemple

en le renommant "*inc_texte-dist.php3*", ce qui permettra de revenir en arrière en remplaçant le fichier "*inc_texte.php3*" en cas de problème.

Ouvrez le fichier "*inc_texte.php3*" dans un éditeur de texte. A la ligne 345 (ligne 375 pour la version 1.8.2 d), vous avez :

```
// Securite : empecher l'execution de code PHP
function interdire_scripts($source) {
    $source = eregi_replace("<(\%|\?|([[:space:]]*)script)", "&lt;\\1",
    $source);
    return $source;
}
```

Voilà le script qui bloque les scripts dans les articles. Je vous propose de le modifier ainsi :

```
// Securite : empecher l'execution de code PHP - Modification: autoriser
javascript a condition
function interdire_scripts($source) {
    $source = eregi_replace("<(\%|\?|([[:space:]]*)script)", "&lt;\\1",
    $source);
    $source = str_replace("<mot-de-passe>", '<script
type="text/javascript" language="JavaScript">', $source);
    $source = str_replace("<mot-de-passe-exterieur>", '<script
type="text/javascript" language="JavaScript" src=', $source);
    return $source;
}
```

En faisant ainsi, on garde la sécurité empêchant un script non autorisé. Pour écrire un script dans un article, au lieu d'écrire au début `<script type="text/javascript" language="JavaScript">` vous écrivez `<mot-de-passe>` (en remplaçant bien sûr par le mot de passe de votre choix). Ce "mot de passe" fait que seuls vous et les rédacteurs à qui vous aurez indiqué ce code pourront utiliser JavaScript dans les articles. Voici un exemple destiné à coder une adresse e-mail :

```
<script type="text/javascript" language="JavaScript">
<!--Begin
user = "votrelogin";
site = "votresite";
document.write('<a href="mailto:' + user + '@' + site + '\>');
document.write(user + '@' + site + '</a>'); // End-->
</script>
```

Ce code fonctionnera si vous écrivez :

```
<mot-de-passe>
<!--Begin
user = "votrelogin";
site = "votresite";
document.write('<a href="mailto:' + user + '@' + site + '\>');
```

```
document.write(user + '@' + site + '</a>');  
// End-->  
</script>
```

De la même façon, si on veut utiliser un fichier externe :

```
<script language="JavaScript" type="text/javascript"  
src="fichier_externe.js"></script>
```

cela devient possible en écrivant :

```
<mot-de-passe-exterieur>"fichier_externe.js"></script>
```

L'utilisation de fichiers externe me parait souvent souhaitable, à moins d'avoir un script très simple. En effet, Spip risque de transformer les caractères "< > [] { et }", et alors le script ne marche plus. On peut évidemment supprimer la mise en forme typographique de Spip, mais je trouve que ce serait "cher payé" pour utiliser JavaScript.

Avertissements

1) Si on modifie le fichier *"inc_texte.php3"* il faut évidemment faire une copie de ce fichier avant d'installer une mise à jour la version de Spip installée. Il faudra modifier le nouveau fichier *"inc_texte.php3"* pour que cette astuce fonctionne à nouveau.

2) Evidemment, modifier un des fichiers de Spip, c'est prendre des risques, nous ne garantissons pas la sécurité de ce procédé.

Et PHP ?

Les astuces présentées ici ont été recherchées parce que le seul article que j'ai trouvé indique un procédé qui ne fonctionne pas.

Par contre, pour inclure un script PHP dans une page Spip, la technique présentée dans la page [Lancer un script PHP dans un article](#) fonctionne bien, et en plus sans se priver de la mise en forme typographique de Spip.

[1] Voir l'article [Zoom d'images avec les CSS](#) sur [OpenWeb](#).

[2] Si ce fichier n'existe pas, il faut le créer en faisant une copie du fichier *"article-dist.html"* que vous renommez *article.html*

[3] A la condition de ne pas utiliser quelque chose commençant par `<script...>` qui serait bloqué car transformé en `<script...>`

[4] Cet exemple est inspiré de [Créer des pop-up intelligentes](#) sur [OpenWeb](#).

[5] Merci à Bertrand Boutilier pour cette astuce. Exemple : l'article [Créer une liste de QCM](#) utilise cette technique.



 [Imprimer cet article](#)

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