

# Du JavaScript dans ma page WordPress

*Je voulais faire une page un peu interactive pour la comparaison de matière. Après moult recherches et tentative de plugins, je me suis décidé à introduire du JavaScript dans ma page WordPress. Au début je me suis dit “pas trop compliqué”, et j’ai pensé éditer, en passant en mode texte, directement dans l’éditeur de WordPress ..... grave erreur !!! ça marche pas !!!!*

*Nota : Pour cet article il est supposé que vous avez quelques vagues idées (pas beaucoup) sur le code [HTML](#) et [JavaScript](#). Si vous n’avez aucun bagage de programmation, je vous conseille de faire un petit tour [ici](#) (pour le HTML) et [ici](#) (pour le JavaScript) avant de continuer votre lecture.*

## *La méthode*

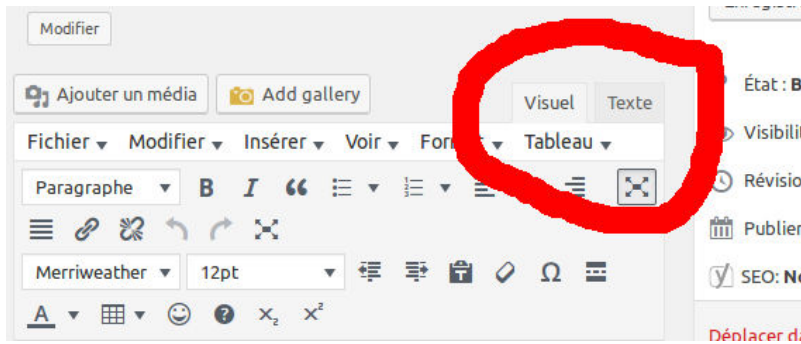
*Je vous propose une méthode, qui a marché pour moi, donc j’espère qu’elle marchera pour vous. Reprenons depuis le début.*

### LES PREMIERS PAS !

*Vous voulez par exemple avoir un bouton sur votre page avec comme objectif de faire apparaître un message lors de son action. Un truc de style ci-dessous, pas vraiment très sophistiqué ☐*

appui pour message

*Ça marche pas !?! Me voilà surpris ☐ mais quel est ce bug ? Si vous écrivez en mode “onglet Texte” dans l’éditeur de WordPress, donc directement en HTML le code pour le bouton, l’éditeur (enfin celui que j’utilise) efface purement et simplement l’appel à la fonction message(). Un simple aller/retour des onglets “Texte-Visuel-Texte” pour vérifier cet état de fait.*



*Écriture en mode texte du code HTML*

*pour insérer un bouton avec la fonction lors du clic :*

```
<p style="text-align: justify;"><em>Vous voulez par exemple  
..... très sophistiqué : </em></p>  
<button onclick="message()">appui pour message</button>  
<p style="text-align: justify;"><em>Le premier problème est que  
.... </em></p>
```

*Retour en mode Texte après un passage en Visuel (clic sur l'onglet "Visuel" et clic sur l'onglet "Texte")*

```
<p style="text-align: justify;"><em>Vous voulez par exemple  
..... très sophistiqué : </em></p>  
<button>appui pour message</button>  
<p style="text-align: justify;"><em>Le premier problème est que  
.... </em></p>
```

*L'appel à la fonction message à disparu ! Vous comprenez tout de suite que votre éditeur WordPress est un problème, mais vous continuez .... ce n'est pas un simple bout de code qui fait la loi, non mais !! ☐*



**ON VA ÉCRIRE L'APPEL EN MODE JAVASCRIPT !**

appui pour message

*Ça ne marche toujours pas ! Car là aussi l'éditeur fait des siennes (c'est pas le seul d'ailleurs au vu des fautes d'orthographe dans le code, déplorable ☐) ! Pour preuve, votre*

*joli code JavaScript entouré des balises HTML adéquates que vous avez délicatement écrit en mode Texte*

```
<script>
document.getElementById('bt').setAttribute('onclick',"message();
");
function message(){
alert("Vous avez appuyé sur le bouton");
}
</script>
```

*Après passage de l'éditeur (clic "Visuel" clic "Texte") ce n'est plus aussi compréhensible. Je n'ai pas compris les règles de modifications, ce doit être selon l'humeur de l'éditeur je suppose*   *(photo d'écran)*

```
<script>
<span style="display: inline-block; width: 0px; overflow: hidden; line-height: 0;" data-mce-type="bookmark"
class="mce_SELRES_start"></span>document.getElementById('bt').setAttribute('onclick',"message();");
function message(){
alert("vous avez appuyez sur le bouton");
}
</script>
```

*Mais peut-être qu'en comprimant, soit en supprimant les espaces et les retours à la ligne, votre code ne sera pas modifié ?*

```
<script>document.getElementById('bt').setAttribute('onclick',"message();");function message(){alert("Vous avez appuyé sur le bouton");}</script>
```

*Non là aussi votre éditeur tente de vous aider !* 


```
<script>document.getElementById('bt').setAttribute('onclick',"message();");function message(){alert("vous avez appuyez sur le bouton");}<span style="display: inline-block; width: 0px; overflow: hidden; line-height: 0;" data-mce-type="bookmark" class="mce_SELRES_start"></span></script>
```

## LA SOLUTION

*Sortir le code JavaScript pour le placer dans un fichier annexe. Pour cela il suffit de placer*

un appel à ce fichier extérieur dans le fichier HTML (l'onglet "Texte" de l'éditeur de WordPress).

appui pour message *celui-là* fonctionne !

Remarquez le petit symbole :  en bas de page, qui vous signale que vous faites appel à un fichier extérieur. Il n'apparaît que dans le mode Édition de WordPress, pas lors de la visualisation des pages de votre site.

*Voilà vous savez tout !*

## Tableau et graphique

Maintenant que la méthode est connue, voici les fichiers utilisés pour réaliser la page [comparer les matières](#). Vous trouverez les fichiers JavaScript et un fichier json en [cliquant ici](#)

Maintenant quelques explications concernant ce code, le pourquoi du comment

### La partie HTML

C'est le code que vous allez écrire directement dans votre éditeur de WordPress, donc maintenant, vous connaissez bien la manœuvre, passage en mode Texte (clic onglet "Texte"). Attention ne copiez pas directement le code ci-dessous, mais prenez le fichier "FichierPage.html".

```
LISTE DES FICHIERS A CHARGER
<!--[if lt IE 9]><script src="excanvas.js"></script><![endif]-->
<script src="flotr2.min.js"></script>
<script type="text/javascript" src="ScriptJS.js"></script>
.....
.....
LISTE DEROULANTE DU NOMBRE DE CARACTERISTIQUES
<form name="NbrCol"><select id="nbrcol" name="Nbr"
```

```

onchange="NbrColJS()"><br>
<option value="0">Toutes</option><br>
<option value="1">1</option>
...
Nombre de caractéristiques à comparer
</select></form>
.....
.....
LISTE DEROULANTE DES CARACTERISTIQUES POSSIBLES
<form name="NameCol"><select id="namecol" name="Name"
onchange="NameColJS()">
<option value="0">Tout</option>
<option value="1">Résistance traction</option>
...
Sélection des caractéristiques
</select></form>
.....
.....
BOUTON ET POSITION DU TABLEAU
<button id="bt">Voir la table</button>
<h2>Tableau</h2>
<div id="tblMat"></div>
.....
.....
POSITION ET TAILLE DU GRAPHIQUE
<h2>Graphique de comparaison</h2>
<div id='chart'
style="width:800px; height:500px;"></div>
...

```

*Attention : supprimer la largeur pour le graphe, problème sur les smartphones.*

*Voilà pour la page HTML, le reste c'est dans les fichiers déclarés au début de cette page soit :*

*excanvas.js* : fichier utile pour les personnes utilisant un navigateur IE9, pour ceux qui utilisent IE8 je ne sais pas si ça marche, je pense que non!

*flotr2.min.js* : C'est le code JavaScript pour la gestion et le traçage du graphique.

*ScriptJS.js* : C'est votre code JavaScript propre (le mien en l'occurrence).

*Les deux premiers fichiers comme vous l'avez certainement compris, proviennent d'internet*

et plus particulièrement d'[ici](#). Que ces contributeurs soient remerciés pour leurs précieux apports. Attention dans vos déclarations d'appel de fichiers, il est nécessaire que votre serveur Web sache les trouver. Donc, copiez ces fichiers à la racine ou mieux dans un répertoire *ah-doc* de votre serveur, dans ce cas n'oubliez pas de modifier leurs chemins d'accès. Je ne parlerais pas de ces deux fichiers car vous trouverez beaucoup d'infos dans le forum [ici](#).

Les listes déroulantes, ce ne sont que des `<form>` à placer où bon vous semble dans le fichier de votre page Web. Je n'ai pas grand chose à dire là dessus. ☐ C'est un peu pareil pour la position du tableau et du graphique qui sont simplement repérés par un "id" d'un `<div>`. N'oubliez pas votre bouton avec son "id" également, rien de bien compliqué.

## Le fichier ScriptJS.js

Vous trouverez dans le fichier du projet, un fichier `ScriptJS-commenter.js`, essayez de regarder ce fichier, j'espère que les informations données sont suffisantes pour comprendre le code qui n'est pas très compliqué en fin de compte.

### Quelques ajouts généraux aux commentaires

1 **Fichier "table.json"**. J'ai souhaité mettre les données dans un fichier séparé pour pouvoir en ajouter ou modifier simplement sans toucher au code. Le choix du "json" c'est simplement la curiosité pour ce type d'encodage. J'ai trouvé un traducteur "json-cvs", (donc lecture sur tableur) qui fonctionne bien, mais pas le contraire "cvs-json". Attention il y a beaucoup de convertisseurs sur le Web et ils sont efficaces mais lorsque je les utilise en entrant le fichier des données puis retour, sans rien toucher, c'est plus le même fichier, il y a plein de modifications, plutôt sur l'aspect du fichier que sur les données. Bref pour l'instant ce choix de fichier, "json", me laisse dubitatif et vraiment songeur. Il va falloir régler ce souci d'ajout de données. ☐

2 Les deux premières fonctions JavaScript servent essentiellement à charger le fichier de données (`table.json`) et former une variable JavaScript "tabMat" avec ces données, le tout après le chargement de la page Web.

```

window.onload = function() {
    .....
    req.open('GET', "https

.....
.....
req.onreadystatechange = function(event) {
    if (this.readyState === XM
.....

```

*3 Le traçage du graphique se trouve dans la partie tri des données, car il est important de connaître la donnée du tri et les valeurs du tri. Le principe est donc : on trie la table puis on relit les valeurs de la caractéristique triées pour en afficher les valeurs en forme graphique.*

```

FONCTION DE TRI DE LA TABLE
function SortTable(sortOn) {
    var table = document.getElem
    .....
    .....
    RECUPERATION DES VALEURS DE LA TABLE TRIEE
    table = document.getElementById('tri');
    tbody = table.getElementsByTagName('tbody')[0];
    rows = tbody.getElementsByTagName('tr');
    var thead = table.getElementsByTagName('thead')[0];

    TRACAGE DU GRAPHE
    (function basic_bars(container) {
    .....
    .....

```

## Conclusion

*Les éditeurs de [CMS \(WordPress\)](#) qui facilitent la production de pages web deviennent un handicap lorsque l'on souhaite avoir une page un peu particulière, plus interactive. Les [plugins](#) (petit programme qui "modifie" le fonctionnement de l'éditeur) devraient eux nous*

*permettre de faire ce que l'on souhaite, mais il faut trouver le bon plugin (s'il existe) et espérer qu'il fonctionne sur la version de votre CMS. ☐*

*Bon courage.*

*Ma situation logicielle*

*Version WordPress : 4.9.4-fr\_FR*

*Thème : Twenty Sixteen Version : 1.4*

*Editeur : TinyMCE Advanced*