

# Le Plugin SPIP minimalJSMP3player

Un mini lecteur MP3 piloté par du javascript



Lecteur de fichier MP3 à interface HTML paramétrable

# 1 Implantation du plugin

Désarchiver le fichier « minimalJSMP3player-spip-plugin.zip » dans le dossier plugins de votre installation SPIP.

L'activer via l'administration des plugins. (Consulter la [documentation officielle](#) pour plus de détails.)

Dans ce plugin, on utilise l'utilitaire javascript d'installation d'objet Flash **swfobject 2.0**, disponible à l'adresse suivante : <http://code.google.com/p/swfobject/>

En conséquence, il conviendra d'intégrer ce fichier dans la partie « head » des pages qui doivent utiliser le plugin:

```
<script type="text/javascript" src="#CHEMIN{javascript/swfobject.js}"></script>
```

*par exemple, si la librairie swfobject est dans le répertoire « /javascript » de votre installation SPIP.*

Le plugin utilise aussi la boîte à outils javascript jQuery (fonctionne à partir de la version 1.1). Celle ci fait partie intégrante de la distribution SPIP.

## 2 La balise #MINIMALJSMP3PLAYER

Lorsque le plugin est activé, on dispose de la balise #MINIMALJSMP3PLAYER qui liste tous les documents « mp3 » joints à l'article, les intègre à la liste des morceaux à jouer (la playlist), et insère dans la page un lecteur mp3 minimal.

Ainsi,

[(#MINIMALJSMP3PLAYER)] implante dans la page un lecteur minimalJSMP3player avec comme interface l'image présente en première page de ce document.

De la même manière on dispose de la balise #MINIMALJSMP3PLAYER\_RUB qui a la même syntaxe que la balise #MINIMALJSMP3PLAYER, mais qui utilise la rubrique du contexte en cours et ajoute à la playlist tous les fichiers MP3 joints aux articles enfants de cette rubrique (on peut donc utiliser cette balise dans un article ou une rubrique).

Il est possible de transmettre des paramètres à ces balises, comme décrit en :

### [6\) Surcharges supplémentaires du comportement](#)

### 3 Les contrôleurs « par chanson »

Lorsque le plugin est activé, il active aussi la détection de contrôleurs « par chanson ».

Ceux ci permettent de jouer (mettre en pause) un des morceau de la playlist.

Par convention, tous les éléments HTML de la classe « psn » sont considéré comme des « contrôleurs par chanson » potentiels. La chanson qu'un contrôleur pilote est indiquée par l'attribut id de l'élément :

```
<a class="psn" id="psn1" href="#">Jouer morceau 1</a>
```

permet de lire/mettre en pause le morceau de rang 1 dans la playlist.

Ainsi, les éléments de la classe « psn » qui ont pour id « psn0 », « psn1 », ... , <psnx> pilotent les morceaux de rang 0,1,..n.

Dans le texte d'un article, on peut insérer un contrôleur par chanson avec le modèle « psn »

```
<psn3|tag=a|text=Jouer morceau 3>
```

Dans un squelette on obtient le même résultat avec la balise « #MODELE{psn} »

```
[(#MODELE{psn}{id=3,tag=a}{text=Jouer morceau 3})]
```

Dans les deux cas d'utilisation du modèle, on insère :  
le son de rang 3, dans une balise <a> avec comme texte « Jouer morceau 3 »

**Les valeurs possibles pour le paramètre "tag" sont "a", "span", "div" et "img".**

Remarque : un seul des paramètres id, tag et text est obligatoire. Les valeurs par défaut respectives sont alors "0", "a" et "écouter"..

Par exemple :

<psn0> et [(#MODELE{psn}{tag=a})] insèrent le contrôleur suivant :

```
<a class="psn" id="psn0" href="#">écouter</a>
```

<psn4> et [(#MODELE{psn}{id=4})] insèrent le contrôleur suivant :

```
<a class="psn" id="psn4" href="#">écouter</a>
```

Dans le cas de l'utilisation de la balise <img>, l'attribut text doit contenir le "chemin/nom" de l'image à utiliser, qui est réputée se trouver dans le sous-dossier "images" du plugin.

Par exemple :

```
<psn2|tag=img|text=minisplay.gif> insère le contrôleur suivant 
```

## 4 Comportement par défaut

Le lecteur **minimalJSMP3player** ne démarre pas la lecture lorsque la page est chargée. L'utilisateur doit cliquer sur un élément « **Play/Pause** » pour démarrer.

Lorsque la lecture du dernier son de la liste est terminée, la tête de lecture est repositionnée sur le début du premier son et le lecteur s'arrête. Pour recommencer la lecture, l'utilisateur doit à nouveau cliquer sur un élément « **Play/Pause** ».

Si l'utilisateur clique sur le contrôle « **Play/Pause** » pendant la lecture, alors le son est mis en pause. La position dans le son est sauvegardée pour reprendre à cette position en cas de nouveau clic sur « **Play/Pause** ».

Si l'utilisateur clique sur le contrôle « **Stop** », la lecture est arrêtée et la tête de lecture est positionnée au début du son en cours.

## 5 Surcharge du comportement par un fichier de configuration

Dans le répertoire du plugin **minimalJSMP3player**, il existe un sous-répertoire « **xml** » dans lequel se situe un fichier nommé « **jsmp3data.xml** ». Ce fichier de configuration est lu et les valeurs qu'il définit viennent surcharger les valeurs par défaut de comportement.

Ce fichier de configuration est constitué comme suit :

```
<?xml version="1.0" encoding="UTF-8"?>
<JSMP3player autoStart="true" loop="false" >
  <sounds>
    <sound url="plugins/minimalJSMP3player/xml/bosco.mp3" />
  </sounds>
</JSMP3player>
```

Les attributs possibles sont :

### - Démarrage de la lecture

**autoStart** : <démarrer automatiquement la lecture ou non (valeurs "true" ou "false");  
Valeur par défaut : false>

### - Lecture en boucle

**loop** : <redémarrer sur le premier morceau de la liste de diffusion lorsque le dernier se termine (valeurs "true" ou "false"); Valeur par défaut : false>

**Pour ne pas modifier la valeur par défaut d'un des attributs, il suffit de l'omettre ou de régler sa valeur à "".**

### - Liste des sons comprise dans une balise <sounds>

**<sound url="<chemin complet vers le son mp3>" />**

**On ajoute autant de ligne qu'il y a de sons par défaut à ajouter à la playlist.**

## 6 Surcharges supplémentaires du comportement

Comme on vient de le voir le lecteur `minimalJSMP3player` a un comportement par défaut qui peut être modifié par un fichier XML. Cela définit un comportement type que l'on choisit d'appliquer pour un contexte précis.

On peut aussi lors de l'implantation communiquer des variables au lecteur qui prennent le pas sur la configuration par défaut et la surcharge par le fichier XML si elle existe. On peut agir sur :

- le démarrage automatique (ou non) de la lecture au chargement de la page avec la variable **autoStart**
- le comportement lorsque le dernier son a été joué (recommencer au début ou arrêter) avec la variable **loop**

comme suit :

```
{autostart= « valeur booléenne »} -> démarrage de la lecture dès la page affichée  
{loop= « valeur booléenne »} -> Lecture de la liste des morceaux en boucle
```

Ainsi,

```
[(#MINIMALJSMP3PLAYER)]
```

*Le lecteur se comporte comme défini dans le fichier xml*

```
[(#MINIMALJSMP3PLAYER{autostart=true})]
```

*La lecture débute automatiquement*

```
[(#MINIMALJSMP3PLAYER{loop=true})]
```

*Le lecteur boucle sur le premier morceau de la liste*

```
[(#MINIMALJSMP3PLAYER{autostart=true}{loop=true})]
```

*La lecture débute automatiquement et boucle sur le premier morceau de la liste*

```
[(#MINIMALJSMP3PLAYER{autostart=false}{loop=false})]
```

*La lecture ne débute pas automatiquement et le lecteur ne boucle pas sur le premier morceau de la liste*

## 7 Modification de l'aspect du lecteur

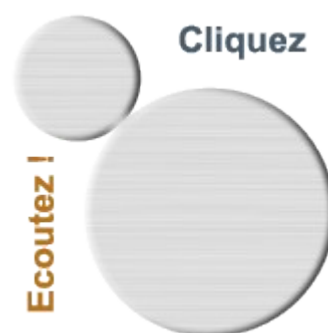
Les éléments graphiques qui composent le lecteur ont pour caractéristiques :

fond : id="interfaceLecteur"

bouton Play/Pause : id="playPauseCtrl"

bouton Stop : id="stopCtrl"

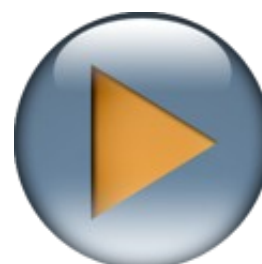
Le positionnement de ces éléments est défini dans le fichier « minimalJSMP3player.css » situé dans le sous-répertoire « css » du plugin. On trouve aussi dans ce fichier la référence à l'image qui compose le fond du lecteur :



Le bouton Stop utilise l'image nommée « images/stop.png »



Le bouton Play utilise l'image nommée « images/play.png »



Le bouton Pause utilise l'image nommée « images/pause.png »

