

## Les balises propres au site

Décembre 2001 – mis à jour le : Avril 2015

Toutes les versions de cet article : [ar](#) [\[ca\]](#) [\[de\]](#) [\[en\]](#) [\[es\]](#) [\[it\]](#) [\[nl\]](#)



Les balises suivantes sont disponibles à n'importe quel endroit du squelette, même en dehors d'une boucle (hors « contexte »).

### Balises définies à la configuration

---

Le contenu de ces balises est défini dans l'espace privé, lors de la configuration de votre site.

- **#NOM\_SITE\_SPIP** affiche le nom du site.
- **#URL\_SITE\_SPIP** affiche l'adresse du site. Elle ne comprend pas le / final, ainsi vous pouvez créer un lien du type `#URL_SITE_SPIP/hello.php`
- **#DESCRIPTIF\_SITE\_SPIP** (depuis **SPIP 1.9**) affiche, comme son nom l'indique, le descriptif du site, que l'on renseigne dans la page de configuration générale du site.
- **#EMAIL\_WEBMASTER** affiche l'adresse du webmestre. Par défaut, SPIP prend l'adresse de celui qui a installé le site (le premier administrateur).
- **#LOGO\_SITE\_SPIP** (depuis **SPIP 1.8**) affiche le logo du site. Depuis **SPIP 1.9** cette balise renvoie le logo du site 0. Il ne faut pas confondre avec le logo de la racine, aussi désigné sous le nom de logo standard des rubriques, c'est-à-dire celui de la rubrique 0.
- **#CHARSET** affiche le jeu de caractères utilisé par le site. Sa valeur par défaut est `iso-8859-1`, jeu de caractères dit « iso-latin » [1].
- **#LANG** : utilisée en dehors des boucles ARTICLES, RUBRIQUES, BREVES et AUTEURS, cette balise affiche la langue principale du site.
- **#LANG\_DIR**, **#LANG\_LEFT**, **#LANG\_RIGHT** (depuis **SPIP 1.7**, **SPIP 1.7.2**) : ces balises définissent le sens d'écriture de la langue du contexte actuel (par exemple, de l'article qu'on est en train d'afficher). Voir l'article « Réaliser un site multilingue » pour plus d'information.
- **#MENU\_LANG** (et **#MENU\_LANG\_ECRIRE**) : ces balises fabriquent et affichent un menu de langues permettant au visiteur d'obtenir la page en cours dans la langue choisie. La première balise affiche la liste des langues du site ; la seconde la liste des langues de l'espace privé (elle est utilisée sur la page de connexion à l'espace privé).

### Balises de mise en page

---

- **SPIP 1.8.2** introduit la balise **#DOSSIER\_SQUELETTE** pour pouvoir développer des squelettes facilement transportables et échangeables. Elle permet d'obtenir le chemin du dossier dans lequel est installé le squelette utilisé.

On peut ainsi placer les fichiers « accessoires » (feuille de style, javascript, etc...) au squelette dans le répertoire du squelette et donc simplement

distribuer ce dossier pour échanger ses squelettes. On écrira donc, par exemple, pour inclure une feuille de style du répertoire squelette :

```
<link rel="stylesheet"
href="#DOSSIER_SQUELETTE/mon_style.css"
```

Depuis **SPIP 1.9**, la balise **#CHEMIN** remplace et améliore **#DOSSIER\_SQUELETTE**. **#CHEMIN{xxx}** donnera le chemin complet vers le fichier XXX, qu'il se trouve à la racine, dans le dossier des squelettes, dans dist/ etc.

```
<link rel="stylesheet"
href="#CHEMIN{mon_style.css}" type="text/css" />
```

- **#PUCE** qui affiche devinez-quoi...

- **#FORMULAIRE\_ADMIN** est une balise optionnelle qui permet de placer les boutons d'administration (« recalculer cette page », etc.) dans ses squelettes. Lorsqu'un administrateur parcourt le site public, si cette balise est présente, elle sera remplacée par les boutons d'administration, sinon, les boutons seront placés à la fin de la page.

Depuis **SPIP 1.8**, on peut aussi modifier la feuille de style *spip\_admin.css* pour contrôler la position des boutons.

- **#DEBUT\_SURLIGNE**, **#FIN\_SURLIGNE** sont deux balises qui indiquent à SPIP dans quelle partie de la page colorer les mots clefs recherchés. Voir : « [Les boucles et balises de recherche](#) ».

- La balise **#INSERT\_HEAD** (depuis **SPIP 1.9.1**) doit se situer entre les balises `<head>` et `</head>` de vos squelettes. Elle permet à SPIP, ainsi qu'aux plugins éventuels, d'ajouter du contenu entre ces deux balises html.

## Balises techniques

---

Attention, ces balises s'adressent à des utilisateurs avertis de SPIP.

- La balise **#REM** ne produit aucun affichage : elle permet de commenter le code des squelettes, de cette façon : `[(#REM) Ceci est un commentaire. ]`. Ces commentaires n'apparaissent pas dans le code généré pour le site public. Voir aussi : [Comment désactiver une partie d'un squelette ?](#)

- **#SELF** (depuis **SPIP 1.8**) retourne l'URL de la page appelée, nettoyée des variables propres à l'exécution de SPIP. Par exemple, pour une page avec l'url : `spip.php?article25&var_mode=recalcul` la balise **#SELF** retournera : `spip.php?article25`

Par exemple pour faire un formulaire :

```
<form action="#SELF" method="get">
```

Remarque : la balise **#SELF** représentant l'adresse de la page, elle n'est pas compatible avec les `<INCLUDE()>` (sauf si le `$delais` de l'inclusion est mis à 0).

- **#URL\_PAGE** (depuis **SPIP 1.9**) retourne une url de type « page » (cf. les urls de spip), vers la page passée en paramètre et qui pourra être utilisée dans un lien. Par exemple, pour accéder à la page générée par le squelette `toto.html`, située dans votre dossier-squelette, **#URL\_PAGE{toto}** générera automatiquement l'url `spip.php?page=toto`. Un second paramètre est autorisé pour ajouter des paramètres à l'url. Exemple

`#URL_PAGE{toto,id_article=#ID_ARTICLE}` générera l'url `spip.php?page=toto&id_article=XXX`.

- `[ (#ENV{xxxx,zzzz}) ]` (depuis **SPIP 1.8**) permet d'accéder à la variable de nom `xxxx` passée par la requête HTTP. `zzzz` est une partie optionnelle qui permet de retourner une valeur même si la variable `xxxx` n'existe pas. On trouve une explication détaillée sur [Spip-Contrib](#)

Par défaut, la balise `#ENV` est filtrée par `htmlspecialchars`. Si on veut avoir le résultat brut, l'étoile « \* » peut être utilisée comme pour les autres balises : `[ (#ENV*{xxxx}) ]`.

Par exemple pour limiter la liste d'auteurs affichés :

```
<BOUCLE_auteurs(AUTEURS) {nom == #ENV{lettre,^A}}>
```

Retourne la liste d'auteur ayant le nom correspondant à l'expression régulière passé dans l'url par la variable `lettre` (`spip.php?page=liste_auteur&lettre=^Z`) ou les auteurs qui ont un nom commençant par un 'A' s'il n'y a pas de variable dans l'url.

- La balise `#SPIP_CRON` (depuis **SPIP 1.8**, n'existe plus en SPIP3) est liée à la gestion par SPIP des calculs qu'il doit faire périodiquement (statistiques, indexation pour le moteur de recherche, syndication de sites etc.).

Si cette balise n'est pas présente sur le site, le moteur de SPIP effectue ses calculs, en temps utile, après avoir envoyé une page à un visiteur ; malheureusement php ne permet pas de fermer la connexion à la fin de la page, et dans certains cas cela peut conduire certains visiteurs malchanceux (ceux dont le passage déclenche une procédure un peu longue, notamment la syndication) à constater une certaine lenteur dans l'affichage de la page demandée.

La balise `#SPIP_CRON` permet de contourner ce problème : son rôle est de générer un marqueur `<div>`

invisible dont la propriété « background » pointe sur le script `spip_background.php` ; ce script à son tour effectue les calculs nécessaires « en tâche de fond », et renvoie une image transparente de 1x1 pixel. Cette astuce permet donc d'éviter tout sentiment de « ralentissement » en déportant les éventuelles lenteurs sur un script annexe.

*A noter* : cette balise n'est pas stratégique, et sa présence ou son absence ne modifient en rien la régularité du calcul des tâches périodiques du site. Depuis SPIP 3.0, cette balise disparaît et n'a plus d'effet dans les squelettes où elle serait encore utilisée.

- La balise `#SET{variable,valeur}` et son pendant `#GET{variable}` ont été introduites par **SPIP 1.9.1**. La balise `#SET{xxx,yyy}` affecte une valeur `yyy` à une variable `xxx propre au squelette calculé`. Cette valeur peut être récupérée par la balise `#GET{xxx}`. Les variables créées ainsi ne sont pas transmises au squelette inclus.

Attention ! Si l'on affecte une valeur à une variable dans la partie facultative avant d'une boucle, il ne sera pas possible de récupérer cette valeur dans la boucle. Cela tient à la manière dont Spip calcule les squelettes.

- La balise `#HTTP_HEADER{argument}` (depuis **SPIP 1.9**) permet de modifier l'entête HTTP de la page retournée par SPIP. Exemple : `#HTTP_HEADER{Content-Type: text/css}`. **Attention !** Le fait d'utiliser cette balise supprime les boutons d'administration. Cette balise

ne peut pas être utilisée dans des squelettes inclus via la syntaxe `<INCLURE>`.

- La balise `#EVAL{argument}` (depuis SPIP 1.9) évalue l'expression PHP mise en accolade. Par exemple `#EVAL{1+1}` affichera 2, `#EVAL{$_DIR_IMG_PACK}` affichera ainsi le chemin vers le répertoire `ecrire/img_pack/`. Attention, il est fortement conseillé de s'en servir avec modération.

- La balise `#CACHE{temps}` permet de déterminer le délai au bout duquel le squelette est réinterprété. Le temps est exprimé en secondes. Il peut se mettre sous forme de calcul. Par exemple : `#CACHE{24*3600}`.

- La balise `#SQUELETTE` existe depuis SPIP 1.9.1, et affiche le chemin du squelette courant.

- La balise `#VAL{argument}` depuis SPIP 2.0 retourne l'argument entre accolade. Par exemple `#VAL{toto}` retourne "toto".

## Notes

---

[1] Cf. [www.uzine.net/article1785.html](http://www.uzine.net/article1785.html) pour une introduction aux charsets, en attendant une documentation plus complète de cette fonctionnalité de SPIP.