

# Le langage SPIP

La base de données de SPIP contient des *tables* (articles, rubriques, auteurs, etc.). A l'intérieur de chacune d'elles, les données sont rangées dans des *champs* (le titre, le descriptif, le chapeau, le texte, etc.).

Fabriquer une page d'un site, nécessite d'extraire d'une table les contenus de certains de ses champs : le titre et le texte d'un article, les titres de tous les articles appartenant à une rubrique, etc.

Ce rôle, habituellement dévolu au PHP et à MySQL, est assuré par le *langage à balises de SPIP*. Avec son système de *boucles*, il gère :

- La syntaxe de la requête de récupération de champs d'une table de la base de données ;
- L'analyse et l'affichage des résultats ;
- Le cas où aucun résultat ne correspond à la requête.

Chaque type de mise en page souhaité (articles, rubriques, auteurs, etc.) est réalisé par un fichier **.html** appelé *squelette*, où le langage SPIP cohabite avec le HTML.

## Les boucles :

```
<Bn>
    Code optionnel avant
<BOUCLEn(TYPE){critère1}{critère2}...{critèreX}>
    Code + balises SPIP
</BOUCLEn>
    Code optionnel après
</Bn>
    Code alternatif
<//Bn>
```

**n**

Identifiant d'une boucle. Débute par '\_' (*underscore*).

Il est bon de choisir un nom explicite ; exemple : ***\_articles\_rubriques***

*Il ne peut pas y avoir deux boucles de même nom dans un fichier squelette.*

**TYPE**

Nom en majuscules de la table explorée par la boucle (exemple ***RUBRIQUES***).

**critère**

Les *critères* sont toujours écrits en minuscules et entre accolades '{' et '}'.

Un critère restreint le nombre de résultats affichés ou indique l'ordre d'affichage. On peut combiner plusieurs critères de sélection successifs ; les résultats sont d'abord triés selon le *critère1*, aux données obtenues on applique ensuite un tri selon le *critère2*, etc.

**balise**

Une *balise* commence par le signe '#' (*dièse*) et s'écrit toujours en majuscules.

Il existe plusieurs types de balises (voir *Les balises*) ; celles désignées par le nom d'un champ de la table spécifiée par *TYPE* donnent accès au contenu du champ portant ce nom (exemple ***#TITRE***).

**code optionnel**

Le *code optionnel* « avant » (précédé de ***<Bn>***) n'est affiché que si la boucle contient au moins une réponse. Il est affiché avant les résultats de la boucle.

Le *code optionnel* « après » (terminé par ***</Bn>***) n'est affiché que si la boucle contient au moins une réponse. Il est affiché après les résultats de la boucle.

Le *code alternatif* (terminé par ***<//Bn>***) s'affiche seul (sans les codes *avant* et *après*) si la boucle ne sélectionne aucun résultat.

## Exemple de boucle :

Afficher tous les titres d'articles de la rubrique définie par la valeur *id\_rubrique* :

```
<B_articles_rubrique>
  Cette rubrique contient les éléments suivants:
<ul>
<BOUCLE_articles_rubrique(ARTICLES){id_rubrique}>
  <li>#TITRE</li>
</BOUCLE_articles_rubrique>
</ul>
</B_articles_rubrique>
  Cette rubrique ne contient pas d'article.
</B_articles_rubrique >
```

## Boucles incluses, boucles successives

Avec des boucles incluses les unes à l'intérieur des autres, chaque boucle incluse dépend du résultat de la boucle qui l'englobe.

Avec des boucles successives, les résultats de chacune d'elles sont indépendants.

*Exemple*, la page d'une rubrique est typiquement constituée des éléments suivants :

```
<BOUCLE_rubrique(RUBRIQUES){id_rubrique}>
  <ul>#TITRE
  <BOUCLE_articles(ARTICLES){id_rubrique}>
    <li>#TITRE</li>
  </BOUCLE_articles>
  <BOUCLE_sous_rubriques(RUBRIQUES){id_rubrique}>
    <li>#TITRE</li>
  </BOUCLE_sous_rubriques>
</ul>
</BOUCLE_rubrique>
  <ul>Il n'y a pas de rubrique à cette adresse.</ul>
</B_rubrique>
```

La première boucle (BOUCLE\_rubrique) dépend de la variable passée dans l'URL de la page (*id\_rubrique=15* par exemple).

Les BOUCLE\_articles et BOUCLE\_sous\_rubriques sont incluses à l'intérieur de la BOUCLE\_rubrique.

- Si la rubrique 15 n'existe pas, la première boucle ne donne aucun résultat (le code alternatif « Il n'y a pas de rubrique... » est affiché), et les deux boucles incluses sont totalement ignorées.
- Si la rubrique 15 existe, les boucles incluses sont analysées. Ces deux boucles sont successives. Elles réagissent en fonction de la première boucle, mais indépendamment l'une de l'autre :
  - ◆ BOUCLE\_articles affichera la liste des articles de la rubrique 15 si elle en contient.
  - ◆ BOUCLE\_sous\_rubriques affichera la liste des sous-rubriques de la rubrique 15 si elle en contient.

## Les critères :

<code>{id_secteur}</code>	Sélectionne les rubriques de ce secteur (secteur = rubrique à la racine du site).
<code>{id_rubrique}</code>	Sélectionne les articles de la rubrique d'identifiant <i>id_rubrique</i> .
<code>{id_parent}</code>	Sélectionne la liste des rubriques contenues dans une rubrique.
<code>{racine}</code>	Sélectionne la liste des secteurs (identique à <code>{id_parent=0}</code> ).
<code>{id_article}</code>	Sélectionne l'article identifié par <i>id_article</i> .
<code>{par hazard}</code>	Donne les résultats en ordre aléatoire.
<code>{par titre}</code>	Classe les résultats par titre en ordre alphabétique.
<code>{par num titre}</code>	Classe selon le numéro, suivi d'un point puis d'un espace, du titre dans l'espace privé. Exemple : <b>05. Manifestations</b> <i>Depuis SPIP 3.0, ce numéro ne paraît pas à l'affichage.</i>
<code>{par date}</code>	Classe par date croissante.
<code>{!par date}</code> ou <code>{par date}{inverse}</code>	Classe par date inverse. Si d'autres critères suivent, seule la date sera en ordre décroissant.
<code>{age comparaison valeur}</code>	Sélectionne les articles par comparaison de leur <i>âge</i> (nombre de jours qui sépare la date de mise en ligne de la date courante) avec le nombre de jours spécifié par <b>valeur</b> .  Exemple lister les articles de moins d'un an dans la rubrique courante : <pre>&lt;B_articles_an&gt; &lt;ul&gt; &lt;BOUCLE_articles_an(ARTICLES){id_rubrique}   {age&lt;365}{par titre}&gt;   &lt;li&gt;#TITRE&lt;/li&gt; &lt;/BOUCLE_articles_an&gt; &lt;/ul&gt; &lt;/B_articles_an&gt;</pre>
<code>{a, b}</code>	Affiche <b>b</b> résultats de la boucle à partir du rang <b>a</b> . <i>Nota : Le 1<sup>er</sup> résultat correspond au rang 0.</i>
<code>{id_rubrique IN x,y}</code>	Limite les résultats aux rubriques <b>x</b> et <b>y</b> .
<code>{!id_rubrique IN x,y}</code>	Exclut les résultats des rubriques <b>x</b> et <b>y</b> .
<code>{!id_rubrique= x }</code>	Exclut les résultats de la rubrique <b>x</b> .
<code>{doublons}</code> ou <code>{unique}</code>	Ces deux critères rigoureusement équivalents évitent les résultats affichés en double.
<code>{logo}</code>	Sélectionne les articles (ou rubriques, etc.) sans logo.

**{statut=xxxxx}**

Sélectionne des articles en fonction de leur statut :

- **{statut=prepa}** en cours de rédaction dans l'espace privé ;
- **{statut=prop}** proposés à la publication ;
- **{statut=publie}** publiés sur le site ;
- **{statut=refuse}** refusés à la publication ;
- **{statut=poubelle}** mis à la poubelle.

**{si condition}**

Le critère **{si ...}** exécute une boucle si la *condition* exprimée en argument est satisfaite.

Exemple :

```
<BOUCLE_art(ARTICLES) {si #ID_RUBRIQUE|=={8}}
{par date}>
  <a href="#URL_ARTICLE">#TITRE</a>
</BOUCLE_art>
```

La boucle s'exécutera uniquement si, dans l'environnement d'exécution de la boucle, **#ID\_RUBRIQUE** est égal à "8".

## Les balises :

- **Issues des tables de la base de données**

<b>#ID_RUBRIQUE</b>	Affiche l'identifiant unique de la rubrique (donné lors de sa création).
<b>#ID_ARTICLE</b>	Affiche l'identifiant unique de l'article (donné lors de sa création).
<b>#TITRE</b>	Affiche le titre (rubrique ou article).
<b>#DESCRIPTIF</b>	Affiche le descriptif (rubrique ou article).
<b>#TEXTE</b>	Affiche le texte (rubrique ou article).
<b>#CHAPO</b>	Affiche le chapeau de l'article.
<b>#PS</b>	Affiche le post-scriptum de l'article.
<b>#ID_SECTEUR</b>	Affiche l'identifiant du secteur dont dépend l'article (le secteur étant la rubrique parente située à la racine du site).
<b>#ID_PARENT</b>	Affiche l'identifiant de la rubrique qui contient la rubrique actuelle. Si la rubrique est située à la racine du site, retourne <b>0</b> .

- **Balises techniques**

**#REM** Permet de commenter le code des squelettes.  
Exemple : **[(#REM) Ceci est un commentaire. ]**  
*Les commentaires ne sont pas transmis en ligne.*

**#SELF** Retourne l'URL de la page appelée, nettoyée des variables propres à l'exécution de SPIP.  
Exemple, pour une page d'URL :  
spip.php?article15&var\_mode=recalcul  
la balise **#SELF** retournera : **spip.php?article15**

<b>#SQUELETTE</b>	Affiche le chemin du squelette courant.
<b>#ENV{variable}</b>	Récupère dans le squelette une variable nécessaire à son exécution et qui lui a été transmise. Par exemple l'id d'un article : http://www.le-pic.org/spip.php?article4 Dans ce cas <b>#ENV{id_article}</b> retourne la valeur 4. <b>id_article</b> fait partie de <i>l'environnement du squelette</i> .
<b>#URL_PAGE</b>	Retourne une URL vers la page passée en paramètre et qui pourra être utilisée dans un lien.  Exemple : <b>#URL_PAGE{toto}</b> retourne spip.php?page=toto pour accéder à la page générée par le squelette toto.html, Autre exemple : <b>#URL_PAGE{toto,id_article=#ID_ARTICLE}</b> retourne spip.php?page=toto&id_article=XXX. <b>#URL_page</b> n'accepte pas plus de 2 paramètres.
<b>#SET{variable, valeur}</b>	Déclare une variable XXX <b>interne au squelette courant</b> et lui affecte la valeur yyy.  <b>Attention</b> : Les variables créées ainsi ne sont pas transmises vers un squelette inclus (voir <i>Les fichiers d'inclusion</i> ).  <b>#SET</b> permet de sauvegarder une valeur déterminée à l'intérieur d'une boucle pour la retrouver par <b>#GET</b> hors de la boucle.
<b>#GET{variable}</b>	<b>#GET{xxx}</b> récupère la valeur affectée à XXX par <b>#SET</b> .
<b>#EXPOSE</b>	Dans une liste ou un menu, met en évidence par un affichage distinctif le lien vers l'élément en cours de lecture.  Exemple d'utilisation dans le squelette article.html : <pre> &lt;BOUCLE_principale(ARTICLES){id_article}&gt;   &lt;B_menu&gt;   &lt;ul&gt;     &lt;BOUCLE_menu(ARTICLES){id_rubrique}&gt;       &lt;li&gt;         &lt;a href="#URL_ARTICLE"[class="#EXPOSE]"&gt;           #TITRE         &lt;a&gt;       &lt;/li&gt;     &lt;/BOUCLE_menu&gt;   &lt;/ul&gt; &lt;/B_menu&gt; #TEXTE &lt;/BOUCLE_principale&gt; </pre> avec, par exemple, les styles suivants : <pre> a { color: blue; } a.on { color: red; font-weight: bold; } </pre>
<b>#AUTORISER</b>	Demande d'autorisation ( <i>modifier, créer, supprimer, etc.</i> ) dans un squelette.  Renvoie un espace si l'autorisation est donnée, et une chaîne vide dans le cas contraire.

**#TOTAL\_BOUCLE** Compte le nombre de résultats affichés par la boucle. Peut s'utiliser dans la boucle, dans ses parties *optionnelles* - « avant » et « après » - ou même dans la partie *alternative* « après » la boucle.

Exemple, afficher le nombre de documents associés à un article :

```
<BOUCLE_art(ARTICLES){id_article}>
  <BOUCLE_doc(DOCUMENTS) {id_article}></BOUCLE_doc>
  [il y a (#TOTAL_BOUCLE) document(s).]
</B_doc>
</BOUCLE_art>
```

**Attention** : Si la partie centrale de la boucle ne retourne rien (cas de <BOUCLE\_doc> qui compte le nombre de résultats) le **#TOTAL\_BOUCLE** ne pourra être affiché que dans la partie *alternative* « après » de la boucle (</B\_doc>).

**#COMPTEUR\_BOUCLE** Retourne le rang de l'itération actuelle de la boucle. On peut par exemple l'utiliser pour numéroter des résultats :

```
<BOUCLE_art(ARTICLES) {par date} {inverse} {0,10}>
  #COMPTEUR_BOUCLE - #TITRE<br>
</BOUCLE_art>
```

**#EVAL{expression}** Evalue l'expression PHP placée entre accolade.

Exemples :

**#EVAL{1+1}** affichera **2**,

**#EVAL{ \_DIR\_IMG\_PACK}** affichera le chemin vers le répertoire écrire/img\_pack/.

*A utiliser avec modération...*

- **Autres**

**#LOGO\_RUBRIQUE** Logo de la rubrique, éventuellement avec la gestion du survol

**#LOGO\_SITE\_SPIP** Logo du site.

**#NOM\_SITE\_SPIP** Affiche le nom du site.

**#URL\_SITE\_SPIP** Affiche l'adresse du site sans le '/' final.

**#URL\_ARTICLE** Affiche l'URL de la page de l'article.

**#URL\_RUBRIQUE** Affiche l'URL de la page de la rubrique.

**#INTRODUCTION{nb caractères}** Affiche le descriptif de l'article limité à *nb caractères*, et à défaut les 600 premiers caractères du début de l'article (chapeau puis texte).

*Les sauts de ligne sont ignorés.*

**#NOTES** Affiche les notes de bas de page (analysées à partir du texte).

**#LESAUTEURS** Affiche les auteurs d'un article, avec lien vers leur propre page publique. Cela évite de créer une boucle AUTEURS pour obtenir le même résultat.

- #DATE**
- A l'intérieur d'une boucle (ARTICLES), (BREVES) ou (RUBRIQUES), la date considérée est la *date de publication* de l'article, de la brève ou la *date de dernière modification* de la rubrique.
  - En-dehors de toute boucle, la date affichée est la date du jour.

**#PETITION** Affiche le texte de la pétition si elle existe. Si elle existe mais que le texte est vide, retourne un espace (une chaîne non vide sans incidence dans une page html).

**#FORMULAIRE\_SIGNATURE** Fabrique et affiche le formulaire permettant de signer la pétition associée à cet article.

**#FORMULAIRE\_FORUM** Fabrique et affiche le formulaire permettant de poster un message répondant à cet article.

**#PARAMETRES\_FORUM** Fabrique et affiche la liste des variables exploitées par le formulaire permettant de répondre à cet article.

## Les filtres

Le comportement et l'affichage des balises est modifiable en leur attribuant des filtres.  
Un filtre commence toujours par ' | ' (*pipe*).

- **Les filtres de #DATE**

**|affdate** **[(#DATE|affdate)]** affiche la date au format texte, exemple « 13 janvier 2020 ».

**|affdate{'Y-m'}** **[(#DATE|affdate{'Y-m'})]** affiche numériquement l'année et le mois de la date filtrée, séparés par un tiret.

**|affdate\_jourcourt** Affiche la valeur numérique du jour, suivie du nom du mois. Exemple : « 6 avril ». Si la date n'est pas dans l'année actuelle, l'année est aussi affichée. Exemple: « 15 novembre 2006 ».

**|affdate\_court** Affiche le numéro du jour suivi du nom du mois. Exemple : « 19 avril ». Si la date n'est pas dans l'année actuelle, alors on affiche le mois et l'année sans le numéros du jour : « Novembre 2004 ».

**|affdate\_mois\_annee** Affiche seulement le mois et l'année : « Avril 2005 », « Mai 2009 ».

**|affdate\_heure** Affiche la date suivie de l'heure : « 3 avril 2005 à 20h00min ».

**|jour** Affiche le jour (nombre).

**|mois** Affiche le mois (nombre).

**|annee** Affiche l'année.

**|heures** Affiche les heures d'une date.

**|minutes** Affiche les minutes d'une date.

secondes	Affiche les secondes.
nom_jour	Affiche le nom du jour (lundi, mardi...).
nom_mois	Affiche le nom du mois (janvier, février...).
saison	Affiche la saison (hiver, été...).

- **Les filtres de texte**

=={valeur}	Vérifie la stricte égalité entre l'élément filtré et <i>valeur</i> (numérique ou alphanumérique avec prise en compte de la casse).
!={valeur}	Vérifie la stricte inégalité entre l'élément filtré et <i>valeur</i> .
>{valeur}	Vérifie si l'élément filtré est strictement supérieur à <i>valeur</i> .
>={valeur}	Vérifie si l'élément filtré est supérieur ou égal à <i>valeur</i> .
<{valeur}	Vérifie si l'élément filtré est strictement inférieur à <i>valeur</i> .
<={valeur}	Vérifie si l'élément filtré est inférieur ou égal à <i>valeur</i> .
couper{nbre de caractères}	<p>Affiche un texte limité au nombre de caractères indiqué entre accolades, ou en l'absence de ce paramètre à 50 caractères.</p> <p>Exemple <code>[ (#TITRE  couper{19}) ]</code></p> <p>Le filtre évite de couper les mots et enlève le formatage du texte. Si le texte est trop long, alors « (...) » est ajouté à la fin.</p>
PtoBR	Transforme les sauts de paragraphe d'un texte en simples passages à la ligne ; (exemple à l'intérieur d'un sommaire).
taille_en_octets	Transforme un nombre d'octets (25678906) en une chaîne de caractères plus explicite (« 24.4 Mo »).
supprimer_tags	<p>Supprime tous les tags html (&lt;...&gt;) mais <i>conserve le contenu</i> de ces mêmes tags ; exemple, pour une balise <b>#DESCRIPTIF</b> contenant :</p> <pre>&lt;p&gt;&lt;strong&gt;William Shakespeare&lt;/strong&gt; naquit à Stratford-sur-Avon, &lt;span class='spip_document_164 spip_documents spip_documents_center' &gt;&lt;img src='IMG/png/maison-2.png' width='68' height='43' alt='' /&gt;&lt;/span&gt; dans une maison sous les tuiles de laquelle était cachée...&lt;/p&gt;</pre> <p><code>[ (#DESCRIPTIF  supprimer_tags) ]</code> retourne :</p> <p>William Shakespeare naquit à Stratford-sur-Avon, dans une maison sous les tuiles de laquelle était cachée...</p> <p><i>Nota : Ce filtre supprime aussi les caractères '&lt;' rencontrés.</i></p>
textebrut	<p>Ce filtre applique les traitements suivants (dans l'ordre) :</p> <ol style="list-style-type: none"> <li>1. remplace une suite d'espaces par un espace simple ;</li> <li>2. remplace les tags &lt;p&gt; et &lt;br&gt; et les nouvelles lignes par un espace simple ;</li> <li>3. supprime tous les tags HTML (voir ci-dessus).</li> </ol>

- **Filtres de test**

`|sinon{argument}` Si ce filtre est appliqué à une balise vide, *argument* est affiché:  
**`[(#TEXTE|sinon{"aucun texte"})]`**  
affichera le contenu de `#TEXTE` s'il existe, et "aucun texte" si `#TEXTE` ne retourne rien.

`!#{sioui,sinon}` **`[(#DESCRIPTIF|?{#DESCRIPTIF,#TEXTE })]`**

Le champ descriptif est-il rempli ?  
*sioui* afficher son contenu,  
*sinon* afficher le contenu du texte.

**Nota :**

*Le sioui est obligatoire et le sinon optionnel.*  
*La virgule est le séparateur **obligatoire** entre le sioui et le sinon.*

`|sinon_interdire_acces` Redirige vers une page d'erreur standard (par défaut sur une 403 dans l'espace privé et 404 dans l'espace public) s'il est appliqué à une variable valant **false**.

Exemple :

**`[(#AUTORISER{webmestre}|sinon_interdire_acces)]`**

La balise `#AUTORISER` vérifie si l'internaute est bien webmestre.  
Le filtre `|sinon_interdire_acces` appliqué au résultat de la balise va faire les redirections idoines si l'utilisateur connecté ne possède pas ces droits.

## Code optionnel

L'utilisation de ce type de syntaxe évite l'affichage de texte(s) précédant et/ou suivant une balise sans contenu...

Exemple :

```
<BOUCLE_articles(ARTICLES){id_rubrique}>  
  #TITRE<br />  
  [Présentation :<br />(#CHAPO)]  
</BOUCLE_articles>
```

Le code optionnel est placé entre crochets.

Il ne s'affiche que si la balise `#CHAPO` placée entre parenthèses renvoie une valeur.

# Les fichiers d'inclusion

Ce sont des fichiers squelettes (*.html*) qui contiennent des parties de code (textes, boucles) communes à plusieurs squelettes :

- les déclarations d'entête des pages HTML (appel des javascript, des feuilles de style...),
- les éléments de navigation communs à la plupart des pages (en haut de page, en colonne de gauche...),
- la bannière supérieure (logo, liens vers les crédits du site, la page de contact...),
- un pied de page...

Les modifications qui concernent plusieurs squelettes sont ainsi facilitées.

Les fichiers d'inclusion s'intègrent tels quels aux squelettes appelants.

Ils sont localisés dans le répertoire *inclure*.

**Syntaxe d'appel :**                    `<INCLUDE{fond=inclure/nom du fichier, paramètre} />`  
Le paramètre (id\_rubrique, ou id\_article, etc.) est *présent si nécessaire*.

**Exemple d'inclusions dans les pages d'un site : le haut et le pied de page**

```
<html>
  <body>
    <div id="page">
      <INCLUDE{fond=inclure/header} />
      .....
      .....
      <INCLUDE{fond=inclure/footer} />
    </div>
  </body>
</html>
```

Exemple de contenu du fichier header.html:

```
<div class="header">
  <div id="logo_site">#LOGO_SITE_SPIP</div>
  <div class="slogan">#SLOGAN_SITE_SPIP</div>
</div>
```

## <INCLUDE> et {doublons}

Si on utilise un **<INCLUDE>** dans un squelette, les doublons mémorisés ne sont pas transmis au squelette inclus.

Ce problème est contourné en rajoutant le paramètre **{doublons}** à l'appel du squelette à inclure.

Exemple                    `<INCLUDE{fond=inclure/mapage, doublons}>`.

**Attention :**                    *les doublons sélectionnés dans le squelette inclus ne "remontent" pas dans le squelette incluant.*