

# Créer mon premier squelette

Avec ce tutoriel, découvrez progressivement le fonctionnement de SPIP en construisant votre premier squelette.

Pas à pas, comment créer un SPIP qui défie les limites.

Créer mon premier squelette.....	1
Mon premier squelette .....	2
Un squelette, plusieurs articles... ..	3
Une rubrique .....	4
Boucles en boucles .....	5
Des filtres.....	6

## Mon premier squelette

(je le sors du placard)

Juin 2001 — maj : Février 2009

Si le système de squelettes peut de prime abord paraître intimidant, c'est que ce qu'on lui demande est suffisamment riche pour l'obliger à être complexe. Mais complexe ne veut pas dire compliqué. Voici un exemple minimal de squelette.

### Matériel requis pour ce tutoriel

- Un SPIP installé quelque part. On supposera, pour commencer, que votre base SPIP contient **au minimum une rubrique et deux articles publiés**. Si ce n'est pas le cas, vous pouvez très vite y remédier en copiant-collant les premiers textes qui vous passent sous la main (vérifiez quand même qu'il ne s'agit pas de votre déclaration enflammée au petit ami de votre voisin de bureau).
- Un éditeur de texte pour créer et modifier les fichiers utilisés par SPIP (par exemple le bloc-notes sous Windows).

1. Dans le dossier `squelettes/` (à créer manuellement s'il n'existe pas à la racine du site), déposez un fichier `tutoriel.html`, qui contient ce qui suit :

```
<BOUCLE_article(ARTICLES){id_article=1}>
#TITRE
</BOUCLE_article>
```

Puis affichez la page `http://votresite.net/spip.php?page=tutoriel` :

SPIP est allé chercher le titre de l'article n°1 de votre base, et l'a inscrit à la place de `#TITRE`.

Si ça ne fonctionne pas, vérifiez que votre article n°1 est bien « publié » (et pas « en attente » ou « en cours de rédaction »).

Puis ajoutez du HTML et d'autres appels de « champs » SPIP, et vous obtenez rapidement votre article n° 1 :

```
<BOUCLE_article(ARTICLES){id_article=1}>
<h1>#TITRE</h1>
<b>#CHAPO</b>
<div align="justify">#TEXTE</div>
</BOUCLE_article>
```

Ajoutez ensuite les champs manquants pour parfaire l'affichage de l'article : `#SURTITRE`, `#LESAUTEURS`, `#SOUSTITRE`, `#NOTES`, etc.

Bien !

## Un squelette, plusieurs articles...

c'est à ça que ça sert !

Juin 2001 — maj : Mars 2008

La leçon précédente nous a permis d'extraire des données de l'article n°1 de la base et d'en faire une page Web. Généralisons...

Notre squelette est bien inutile s'il ne sert qu'à afficher l'article n°1. **Apprenons-lui à afficher n'importe quel article :**

Pour cela nous allons appeler notre page Web avec un paramètre, du type `id_article=2` : dirigez votre navigateur sur l'URL suivante :

« [http://votresite.net/spip.php?page=tutoriel&id\\_article=2](http://votresite.net/spip.php?page=tutoriel&id_article=2) »

S'affiche... toujours l'article 1 (et pas 2). Modifions dans le squelette `tutoriel.html` la ligne qui définit la « boucle article » :

```
<BOUCLE_article(ARTICLES){id_article}>
```

Comme vous le voyez, on remplace simplement `{id_article=1}` par `{id_article}` tout court.

Voilà : [http://votresite.net/spip.php?page=tutoriel&id\\_article=2](http://votresite.net/spip.php?page=tutoriel&id_article=2) vous donne maintenant l'article 2. (Non ? Il devrait...)

La `BOUCLE_article` s'exécute dans un « contexte » où `id_article` est égal à 2 (c'est la valeur qui est passée dans l'URL). Si on lui précise `{id_article=1}` elle va chercher l'article n° 1, mais si on lui demande juste `{id_article}`, elle va chercher l'article dont le numéro est indiqué par le contexte (ici l'URL).

Visitez maintenant ces pages :

- [http://votresite.net/spip.php?page=tutoriel&id\\_article=1](http://votresite.net/spip.php?page=tutoriel&id_article=1),
- [http://votresite.net/spip.php?page=tutoriel&id\\_article=2](http://votresite.net/spip.php?page=tutoriel&id_article=2) et
- <http://votresite.net/spip.php?page=tutoriel>.

Voyez-vous la différence ? Les deux premières pages vous donnent les articles n°1 et 2, la troisième n'a pas d'`id_article` dans son contexte, et génère une erreur.

NB : N'oubliez pas de recalculer la page (bouton en haut à droite de votre page) pour prendre en compte les modifications de votre squelette.

Bravo ! Votre squelette est maintenant « contextuel ».

## Une rubrique

ou comment faire des listes du contenu de la base

Juin 2001 — maj : Janvier 2007

La leçon précédente nous a appris à afficher des éléments en fonction du contexte. Nous allons ici voir comment ce contexte varie au fur et à mesure des BOUCLES rencontrées.

Modifions notre squelette « `tutoriel.html` » de la manière suivante :

```
<BOUCLE_article(ARTICLES)>
#TITRE<br />
</BOUCLE_article>
```

Là, on supprime carrément la condition `{id_article}`. Attention : cette BOUCLE peut générer une page énorme si votre base contient déjà pas mal d'articles : mieux vaut prendre nos précautions et ajouter tout de suite `{0,10}` pour limiter aux 10 premiers articles...

```
<BOUCLE_article(ARTICLES){0,10}>
```

Résultat : en appelant simplement `http://votresite.net/spip.php?page=tutoriel` (plus besoin d'`id_article` désormais, puisque cette condition a été supprimée) les titres des 10 premiers articles publiés s'affichent, séparés chacun par un saut de ligne. À partir de là, on voit comment on peut produire le sommaire d'une rubrique : affichons les 10 articles les plus récents appartenant à cette rubrique.

```
<BOUCLE_article(ARTICLES){id_rubrique}{par
date}{inverse}{0,10}>
<a href="#URL_ARTICLE">#TITRE</a><br>
</BOUCLE_article>
```

Prenons dans l'ordre :

- **id\_rubrique** : ne prend que les articles appartenant à la rubrique `id_rubrique` (cf. ci-dessous pour que cette variable soit définie dans le contexte de notre BOUCLE\_article);
- **{par date}{inverse}** : trie par date dans l'ordre chronologique décroissant...
- **{0,10}** : ... et prend les 10 premiers résultats.
- Enfin, `<a href="#URL_ARTICLE">#TITRE</a>` va afficher non seulement le titre de l'article mais en plus créer un lien vers cet article.

Reste à invoquer le squelette, en lui passant le contexte `id_rubrique=1` :

```
http://votresite.net/spip.php?page=tutoriel&id_rubrique=1
```

La magie de SPIP tient dans la combinaison de ce type de fonctionnalités. Si vous êtes arrivé jusqu'ici, c'est gagné !

## Boucles en boucles

plusieurs niveaux de lecture

Juin 2001 — maj : Décembre 2007

Nous savons générer une liste de titres dans une rubrique. Maintenant, nous allons afficher, sur la même page, les éléments de la rubrique elle-même : son titre, son texte de présentation, etc.

Essayez !

Et voici une solution :

```
<BOUCLE_rubrique(RUBRIQUES){id_rubrique}>
<h1>#TITRE</h1>

<BOUCLE_article(ARTICLES){id_rubrique}{par date}{inverse}{0,10}>
<a href="#URL_ARTICLE">#TITRE</a><br/>
</BOUCLE_article>

[(#TEXTE|justifier)]
</BOUCLE_rubrique>
```

On appelle la page avec

[http://votresite.net/spip.php?page=tutoriel&id\\_rubrique=1](http://votresite.net/spip.php?page=tutoriel&id_rubrique=1).

Que s'est-il passé ici ?

Notre boucle `ARTICLES` est intégrée dans une boucle `RUBRIQUES`. Le contexte de la boucle `ARTICLES` est l'`id_rubrique` donné par la boucle `RUBRIQUES`, qui elle-même va chercher le contexte donné par l'URL (`id_rubrique=1`). Donc nous sommes bien, au niveau des `ARTICLES`, avec l'`id_rubrique` demandé. De ce point de vue rien ne change.

En revanche, la boucle `RUBRIQUES` a permis à SPIP de sélectionner les valeurs des champs de la rubrique en question : on peut donc afficher le `#TITRE` et le `#TEXTE` de cette rubrique. Notez bien que ce `#TEXTE` serait celui de la rubrique *même si* on appelait aussi `#TEXTE` dans la boucle `ARTICLES`. Le fonctionnement arborescent de SPIP garantit que le `#TEXTE` d'un article ne déborde pas de la boucle `ARTICLES`...

Dernière remarque : on a introduit un *filtre* `|justifier` sur le champ `#TEXTE`. Ce filtre modifie le contenu du texte avant de l'installer dans la page finale. Ca vous fait saliver ?

## Des filtres

Subtilités squelettiques

Juin 2001 — maj : Octobre 2003

*Si les BOUCLES permettent de structurer la page de manière logique, reste à présenter les données de manière esthétique. Question dizahigne SPIP ne peut rien pour vous, mais sachez user de ses philtres...*

Une donnée stockée dans la base de données se présente comme un bloc de texte, et on peut avoir envie de manipuler sa valeur avant de l'afficher à l'écran. Les *filtres* sont faits pour ça :

- les filtres les plus utilisés (ils sont appelés automatiquement) sont `|typo` et `|propre` ; le premier est un correcteur typographique, dont la mission principale est d'ajouter des espaces insécables où il en faut (cf. l'aide en ligne de SPIP) ; le second s'intéresse aux paragraphes, aux raccourcis SPIP (italiques, gras, intertitres, etc.) - il n'est appliqué par défaut qu'aux textes longs (`#TEXTE`, `#CHAPO`, etc.)
- d'autres filtres sont très utiles : citons `|majuscules` (à la fonctionnalité évidente), `|justifier` ou `|aligner_droite` (qui définissent l'alignement du texte par rapport aux bords verticaux), ou encore l'ésotérique `|saison` (qui affiche « été » si la variable est une date comprise entre le 21 juin et le 20 septembre)...

Pour utiliser un filtre il faut entourer la variable de parenthèses et de crochets (on verra plus tard les implications) : `[blah blah (#VARIABLE|filtre) bloh bloh]`

On peut enchaîner les filtres les uns à la suite des autres [\[1\]](#) : ainsi `[(#DATE|saison|majuscules)]` affichera-t-il « HIVER ».

*Exercice portant sur l'ensemble des leçons précédentes : Afficher en majuscules les titres des 10 articles les plus récents de la rubrique passée en contexte, et mettre en tête de page la saison courante (c'est-à-dire la saison à laquelle a été publié l'article le plus récent de toute la base).*

...

**Pourquoi ces crochets ?** Supposons que votre base contient des articles datés et d'autres non datés. La variable `#DATE` vaut « 2001-07-01 10-53-01 » (date au format mySQL) dans le premier cas, et « 0000-00-00 00-00-00 » dans le second. Pour afficher la date dans un joli (?) cadre, on va utiliser, dans le squelette, les lignes suivantes :

```
[<table border="1"><tr><td>
(#DATE|affdate)
</td></tr></table>]
```

Ici le filtre `|affdate` affiche la date en lettres (au format « 1er juillet 2001 »), mais renvoie une chaîne vide si la date est inconnue (égale à « 0000... »). Les crochets délimitent ce qu'il faut afficher autour de la date *si le resultat entre parenthèses n'est pas une chaîne vide*.

Résultat : seuls les articles datés provoquent l’affichage d’un tableau contenant la date. Un squelette bien construit définira précisément ce qu’il faut afficher *ou pas* en fonction du contenu... Les filtres servent aussi à ça.

## **P.-S.**

Notons que certains filtres de présentation peuvent être avantageusement remplacés par des feuilles de style. Ainsi `|majuscules` est équivalent à l’attribut CSS « `text-transform: uppercase` », et `|justifier` à « `text-align: justify` ».

Lire « Les feuilles de style de SPIP » pour plus de détails sur les styles CSS offerts par SPIP.

## **Notes**

[1] On peut appeler ça un « *pipeline* »...