

Leçon 4. Réaliser une boucle SPIP

Objectifs

Au terme de cette leçon vous devrez être capable de:

1. Créer vos propres boucles

Objectif

La création des boucles est primordiale pour réaliser un squelette car sans elles, vous ne pourriez pas récupérer les données dans la base et les mettre en ligne.

1. Afficher un article

Objectif

On veut décrire de façon générique la façon dont sera affiché un article sur le site public (habillage graphique, éléments affichés, navigation...). On veut :

- le décrire en HTML (comme pour une page normale)
- que la description s'adapte automatiquement à tous les articles de la base de données

Code html

Par exemple on aimerait le code HTML suivant (volontairement simpliste) :

```
<html>
<head>
  <titre>Le titre de l'article</titre>
</head>

<body>
  Le texte de l'article
</body>
</html>
```

Il faut tout d'abord créer le fichier "**article.html**" dans un nouveau répertoire que vous créez à la racine et que vous appelez **squelettes**. Ce fichier décrit la mise en page de tous les articles sur le site public.

Récupérer les champs Il faut indiquer à SPIP quels éléments il doit afficher dans la page :

```
<html>
<head>
  <titre>#TITRE</titre>
</head>

<body>
  #TEXTE
</body>
</html>
```

Les "**balises**" de type "**#TEXTE**" (le dièse et les majuscules sont obligatoires) indiquent à SPIP qu'il faut tirer un contenu de la base de données et l'insérer dans la page HTML, à l'endroit donné.

Pour quel article

Il reste un problème : notre description doit être générique, or rien ne dit à SPIP de quel article on veut afficher le "**#TITRE**" et le "**#TEXTE**"... Il faut une structure supplémentaire :

```
<BOUCLE_principale(ARTICLES) {id_article}
<html>
<head>
  <titre>#TITRE</titre>
</head>

<body>
  #TEXTE
</body>
</html>
</BOUCLE_principale>
```

Décomposition de ce que nous venons d'écrire

- pour indiquer quel(s) article(s) chercher dans la base de données, il faut une "boucle"
- cette boucle est constituée d'un tag ouvrant et d'un tag fermant :
<BOUCLE_principale ... > ... </BOUCLE_principale>
- le tag ouvrant contient deux informations :
 - le nom de la table dont la boucle dépend est entre parenthèses (ici une boucle **ARTICLES**),
 - le critère de sélection (ici "**id_article**")
- la boucle porte un nom choisi librement, mais commençant par "**BOUCLE_**" : ici c'est "**BOUCLE_principale**".

Vérification en ligne

Une fois la boucle recopiée et sauvegardée dans le fichier "**article.html**", on peut afficher un article dans le site public et constater que notre mise en page est effectivement appliquée. (par exemple, choisissez un article publié dans l'espace privé et cliquez sur "voir en ligne")

Explication du résultat :

- l'URL appelée est de la forme : `article.php3?id_article=12`
- "**article.php3**" va chercher notre fichier "**article.html**"
- "**id_article=12**" fait coïncider le critère "**id_article**" de la "**BOUCLE_principale**" avec l'article portant le n°12 : alors **#TITRE** et **#TEXTE** sont remplacés, respectivement, par le titre et le texte de cet article.

Important

la boucle "**ARTICLES**" ne sélectionne que les articles publiés. On ne peut pas afficher les autres articles sur le site public. De même si une rubrique n'a pas d'articles publiés, elle n'apparaîtra pas.

Le fichier "**article.html**" ainsi créé est le squelette de mise en page correspondant aux articles.

Dans le manuel : la syntaxe des boucles
http://www.spip.net/fr_article898.html

2. Les champs d'un article

Les balises

Toutes les balises d'un article sont récupérables dans un squelette, par exemple : **#SURTITRE**, **#TITRE**, **#SOUSTITRE**, **#CHAPO**, **#TEXTE**, **#PS** (pour les post-scriptum), **#NOTES** (affiche automatiquement les notes en bas de page),

Exemple

```
<BOUCLE_principale(ARTICLES) {id_article}
<html>
  <head>
    <titre>#TITRE</titre>
  </head>

  <body>
    <h2>#SURTITRE</h2>
    <h1>#TITRE</h1>
    <h2>#SOUSTITRE</h2>
    <strong>#CHAPO</strong>
    <div align="justify">#TEXTE</div>
    <blockquote>Post-scriptum : #PS </blockquote>
  </body>
</html>
</BOUCLE_principale>
```

Dans le manuel : la boucle ARTICLES

http://www.spip.net/fr_article902.html

Comment éviter d'afficher du texte si un champ est vide (par exemple le "**#PS**") :

```
[...]
[<blockquote>Post-scriptum : (#PS) </blockquote>]
[...]
```

Décomposition de cette syntaxe avancée :

- Le champs **#PS** est mis entre parenthèses pour le distinguer du texte autour.
- le texte "conditionné" à l'existence du champ **#PS** est mis entre crochets : il n'est affiché que si le champ **#PS** (c'est-à-dire le post-scriptum de l'article) n'est pas vide

3. Un sommaire : les dix derniers articles publiés

Accueil du site public Pour modifier l'apparence de la page d'accueil du site public, il faut créer le fichier "**sommaire.html**".

Les articles On veut afficher les derniers articles publiés, sur le modèle :

```
<html>
<body>
  <h2>"#TITRE" du dernier article publié</h2>
  <h2>"#TITRE" de l'avant-dernier article publié</h2>
  .....
...</body>
</html>
```

Pour dire à SPIP d'afficher ces articles, il faut

- une boucle ARTICLES
- un critère pour récupérer tous les articles publiés
- un critère pour classer par ordre inverse de date de publication

```
<html>
<body>
  <BOUCLE_sommaire(ARTICLE) {tout} {par date} {inverse}>
    <h2>#TITRE</h2>
    <a href="#URL_ARTICLE">Lire l'article</a>
  </BOUCLE_sommaire>
...</body>
</html>
```

Décomposition :

- **{par date}** signifie qu'il faut trier par ordre chronologique
- **{inverse}** inverse l'ordre de tri, afin d'aller du plus récent au plus ancien
- la boucle est affichée autant de fois qu'il y a d'articles (d'où le terme de "**boucle**")

Les dix derniers

Pour se restreindre aux dix derniers articles publiés, il faut ajouter un critère supplémentaire disant au système de s'arrêter après les dix premiers résultats :

```
<BOUCLE_sommaire(ARTICLE) {tout} {par date} {inverse} {0,10}
```

Décomposition : dans **{0,10}**, le deuxième nombre est le nombre de résultats à afficher, le premier nombre est le numéro du premier résultat à afficher (dans l'ordre de tri). En informatique, les numérotations commencent à zéro (et non à un). On affiche donc ici dix articles à partir du premier (qui porte le numéro zéro).

4. Auteur d'un article : des boucles imbriquées

Objectif

On veut afficher les auteurs de chaque article. On utilise pour cela une boucle de type "AUTEURS" à l'intérieur de la boucle "ARTICLES".

Tous les auteurs

Si on se contente d'une boucle sans critère, on affiche... tous les auteurs du site.

```
<html>
<body>
<BOUCLE_sommaire(ARTICLE) {tout} {par date} {inverse}>
<h2>#TITRE</h2>
<a href="#URL_ARTICLE">Lire l'article</a>
<p>Article écrit par :
  <BOUCLE_auteurs(AUTEURS)>
    #NOM
  </BOUCLE_auteurs>
</p>
</BOUCLE_sommaire>
...</body>
</html>
```

Pour n'afficher que les auteurs de l'article, on ajoute un critère **{id_article}** à la boucle AUTEURS, pour indiquer que l'on affiche bien les auteurs de l'article en question.

```
<html>
<body>
<BOUCLE_sommaire(ARTICLE) {tout} {par date} {inverse}>
<h2>#TITRE</h2>
<a href="#URL_ARTICLE">Lire l'article</a>
<p>Article écrit par :
  <BOUCLE_auteurs(AUTEURS) {id_article}>
    #NOM
  </BOUCLE_auteurs>
</p>
</BOUCLE_sommaire>
...</body>
</html>
```

Décomposition de l'exemple : la boucle AUTEURS sélectionne les auteurs de l'article courant, c'est-à-dire l'article affiché par la boucle ARTICLES englobante

Affinement

ne pas afficher le tiret en trop à la fin de la liste des auteurs, et
ne rien afficher si l'article est anonyme

```
[ ... ]
.....<B_auteurs>
  <p>Article écrit par :
    <BOUCLE_auteurs(AUTEURS) {id_article}>
      #NOM
    </BOUCLE_auteurs>
  </p>
.....</B_auteurs>
[ ... ]
```

Décomposition :

- Ce qui est entre <B_auteurs> et la BOUCLE_auteurs s'affiche que si la boucle *n'est pas vide*
- Ce qui est entre BOUCLE_auteurs et la </B_auteurs> s'affiche que si la boucle *n'est pas vide*
- ceci s'adapte à tous les types de boucles !

Dans le manuel : la boucle AUTEURS

http://www.spip.net/fr_article907.html

5. Un sommaire par rubrique : les boucle récursives

Objectif On veut maintenant afficher les articles par rubrique.

Articles dans rubriques On utilise donc une boucle **RUBRIQUES**, avec une boucle **ARTICLES** imbriquée

```
<html>
<body>
  <BOUCLE_rubriques(RUBRIQUES) {par titre}>
    <h2>#TITRE</h2>

    <B_articles>
    <ul>
      <BOUCLE_articles(ARTICLES) {id_rubrique} {par titre}>
        <li><a href="#URL_ARTICLE">#TITRE</a></li>
      </BOUCLE_articles>
    </ul>
    </B_articles>
    Pas d'article dans cette rubriques </p>
  </B_articles>

  <BOUCLE_rubriques>
</body>
</html>
```

Rubriques imbriquées Cependant les rubriques s'affichent ici à plat, et non sous la forme arborescente souhaitée. Il faut donc tout d'abord se limiter aux rubriques de premier niveau ("à la racine" du site), en ajoutant le critère approprié. Ensuite, une deuxième boucle **RUBRIQUES** imbriquée dans la première permet d'afficher les sous-rubriques de manière imbriquée.

```
<html>
<body>
  <BOUCLE_rubriques(RUBRIQUES) {racine} {par titre}>
    <h2>#TITRE</h2>

    <B_sous_rubriques>
    <ul>
      <BOUCLE_sous_rubriques(RUBRIQUES) {id_parent} {par titre}>
        <li><strong>#TITRE</strong></li>
      </BOUCLE_sous_rubriques s>
    </ul>
    </B_sous_rubriques>

  <BOUCLE_rubriques>
</body>
</html>
```


Niveaux d'imbrication

Problème : on veut afficher toute l'arborescence des rubriques, mais on ne sait pas à l'avance combien de niveaux d'imbrication il faut prévoir dans le squelette...

Solution : on utilise une boucle réursive, c'est-à-dire une boucle qui "bouclera" sur une autre tant qu'il y a de nouvelles choses à afficher.

```
<html>
<body>
  <BOUCLE_rubriques(RUBRIQUES) {racine} {par titre}>
  <h2>#TITRE</h2>

  <B_sous_rubriques>
  <ul>
    <BOUCLE_sous_rubriques(RUBRIQUES) {id_parent} {par titre}>
    <li><strong>#TITRE</strong></li>
    <BOUCLE_recursive(BOUCLE_sous_rubriques)>
    </BOUCLE_recursive>
  </BOUCLE_sous_rubriques s>
  </ul>
  </B_sous_rubriques >

  <BOUCLE_rubriques>
</body>
</html>
```

Les boucles récursives sont utiles pour les structures arborescentes illimitées, c'est-à-dire : les rubriques (boucle RUBRIQUES) et les forums (boucle FORUMS).

Dans le manuel :

la boucle RUBRIQUES : http://www.spip.net/fr_article904.html

les boucles récursives : http://www.spip.net/fr_article914.html

6. Afficher les brèves

Objectif On utilise ici la boucle BREVES. Elle sélectionne automatiquement les brèves publiées.

Dix dernières Par exemple pour afficher les dix dernières brèves publiées sur tout le site :

```
<BOUCLE breves(BREVES) {par date} {inverse} {0,10}>
```

Dans un rubrique Les dix dernières brèves publiées dans une rubrique particulière:

```
<BOUCLE breves(BREVES){id_rubrique} {par date} {inverse} {0,10}>
```

Le squelette d'une brève peut être modifié en créant le fichier "**breves.html**" :

```
<BOUCLE_principale(BREVES) {id_breve}>
<html>
<head><titre>#TITRE</titre></head>
<body>
<h1>#TITRE</h1>
<div align="justify">#TEXTE</div>
[<div align="right">Voir en ligne :
<a href="#URL_SITE">#NOM_SITE</a></div>]
</body>
</html>
</BOUCLE_principale>
```

Dans le manuel : la boucle BREVES
http://www.spip.net/fr_article906.html

7. Résumons

Les boucles

Voici la structure d'une boucle

```
<B_nomdelaboucle>  
  Code HTML optionnel avant  
<BOUCLE_nomdelaboucle(TYPE) {critère1} {critère} ...{critèreX}>  
  code HTML + balises SPIP  
</BOUCLE_nomdelaboucle>  
  code HTML optionnel après  
</B_nomdelaboucle>  
  Code HTML alternatif  
<\/B_nomdelaboucle>
```

Les balises

se placent à l'intérieur d'une boucle et est toujours précédé du signe dièse "#". Chaque type de boucle a ses balises.

Les critères

indiquent à la fois selon quels critères on veut sélectionner les éléments de la base de données et la façon dont on va les classer ou sélectionner les éléments.

8. Propos d'étapes

Nous venons de voir comment réaliser une boucle dans une page HTML

Nous avons vu les principes de base de SPIP (leçon 1), l'installation du logiciel en ligne (leçon 2), la structure de SPIP (leçon 3) et la réalisation des boucles (leçon 4).

Il nous reste à voir où on peut trouver de l'aide (leçon 5)