



Manuel de « référence »

Version hors-ligne du site Spip.net

Tome 4

Questions et réponses
Le développement de SPIP et ses outils
Traduire SPIP

SPIP est un système de publication pour l'Internet qui s'attache particulièrement au fonctionnement collectif, au multilinguisme et à la facilité d'emploi. C'est un logiciel libre, distribué sous la licence GNU/GPL. Il peut ainsi être utilisé pour tout site Internet, qu'il soit associatif ou institutionnel, personnel ou marchand.

SPIP est développé (programmé, documenté, traduit, etc.) et utilisé par une communauté de personnes que chacun est invité à rejoindre (ou simplement à contacter) sur différents sites Web, listes de discussion par email et rencontres (les fameux « Apéros-SPIP »). Le programme est né en 2001 d'une initiative du minirézo, un collectif défendant le Web indépendant et la liberté d'expression sur Internet. Il est actuellement utilisé sur des dizaines de milliers de sites très divers. Ce site est la documentation officielle.

Le document que vous avez entre les mains ou sur votre écran est un copier/coller du site Spip.net effectué entre le 10 et le 13 septembre 2007.

Ce document est destiné à ceux qui n'ont pas de connexion internet ou qui préfèrent avoir une version papier. Mais si vous voulez avoir la dernière version en ligne et à jour rendez vous sur le site Spip.net.

Je n'est fait aucune modification ou correction par rapport au site, j'ai juste revu la mise en page afin d'adapter le site sur « papier ».

Ce document a été réalisé avec [OpenOffice.org 2.2.1](http://OpenOffice.org) et la police « [Libération](#) » de RedHat.

Bonne lecture. Fabien.

P.S.: La plupart des liens hypertexte ont été laissés afin d'avoir accès aux articles en lignes.

Questions et réponses

Comment obtenir des réponses ?

Les espaces d'entraide entre utilisateurs et utilisatrices de SPIP sont nombreux et très actifs. Chacun est donc invité à faire son possible pour ne pas les surcharger inutilement : les personnes qui interviennent le font à titre bénévole et ne répondront pas à des demandes trop pressantes ou déplacées.

Avant tout, commencez par bien consulter la présente documentation, notamment les qui y figurent ; beaucoup de questions y sont abordées. De nombreux utilisateurs de SPIP tiennent à jour [le site des contributions externes SPIP-CONTRIB](#) : c'est une mine d'informations et de solutions à des problèmes variés.

Une dernière recommandation : *ne multipliez pas inutilement les appels à l'aide* en postant le même message en différents endroits (listes de discussion, forums...). C'est le meilleur moyen de vous faire mal voir des personnes qui auraient pu vous apporter de l'aide.

Liens utiles

- [La liste des utilisateurs de SPIP : spip@rezo.net](#)

Cette **liste de discussion** est destinée à toutes les questions autour de l'utilisation de SPIP. C'est l'endroit le plus pertinent pour obtenir de l'aide.

Il s'agit donc de notre liste principale : si vous êtes webmestre d'un site SPIP, rédacteur d'un site SPIP, et que vous voulez discuter de différents problèmes liés à l'*utilisation* de SPIP, c'est là qu'il faut vous rendre...

Si vous débutez avec SPIP, nous vous conseillons vivement de vous abonner à cette *liste des utilisateurs* (tout cela est évidemment gratuit). Cette liste est très animée, réactive, et vous y obtiendrez rapidement de nombreux conseils et des réponses à toutes sortes de questions.

Attention : Cette liste est très active. Avant de poster, merci de consulter ses [archives](#) publiées en ligne et que l'on peut interroger grâce à un moteur de recherche, pour voir si le sujet n'a pas déjà été abordé. Merci également de consulter la présente documentation (en particulier les FAQ).

- [SPIP Forums](#)

Le site « SPIP Forums » vous propose un **forum de discussion**. Il est recommandé pour *les utilisateurs occasionnels*. Comme sur tout forum, merci de consulter les messages déjà postés pour voir si votre question n'a pas déjà été posée précédemment et la réponse publiée.

- [Canal IRC](#)

Canal de discussion en direct sur [IRC](#), où vous pouvez rencontrer des utilisateurs de SPIP. Si vous ne disposez pas de logiciel IRC, vous pouvez utiliser [ChatZilla](#), libre et gratuit !

- [SPIP Party](#)

Depuis novembre 2005, le site « SPIP Party » rend compte, ville par ville, des **rencontres** qui s'organisent entre toutes les personnes intéressées par la publication sur le Web avec SPIP.

- [Aide à la logistique](#)
- [Aide à la logistique : Covoiturage](#)
- [Apéro Lille - saison 2007-2008](#)

Questions et réponses

• FAQ Premiers pas

- [Lors de l'installation, la vérification des droits d'accès échoue à cause des répertoires CACHE et IMG. Pourtant, je suis sûr d'avoir réglé les droits correctement.](#)
- [Mon site sous SPIP est planté, est-ce que je vous fais un procès ?](#)
- [Je voudrais une nouvelle fonctionnalité...](#)
- [Est-ce que je peux vendre un service utilisant SPIP ? Est-ce que je peux monter une activité professionnelle autour de SPIP ?](#)
- [Est-ce que je peux utiliser SPIP pour réaliser un site marchand, un site avec des pubs, ou un site de droite, ou un site qui débîne le logiciel libre...](#)
- [Comment réinstaller SPIP si je n'arrive plus à accéder à l'espace privé ?](#)
- [De quoi a besoin SPIP pour fonctionner ?](#)

• FAQ technique

- [Ajouter un type de document](#)
- [Des liens qui ouvrent une nouvelle fenêtre](#)
- [Je suis chez Free ou chez Online, et j'obtiens souvent des messages « Maximum execution time exceeded »](#)
- [Pourquoi la création automatique de vignettes ne fonctionne pas avec tous les formats d'images \(JPEG, GIF, PNG\) ?](#)
- [Comment fonctionne le moteur de recherche de SPIP ?](#)
- [Est-ce qu'il faut réinstaller une sauvegarde de la base de données après avoir fait une mise à jour de SPIP ?](#)
- [Je viens d'effectuer une mise à jour de SPIP, et le site public fonctionne mal](#)
- [Après modification de mes squelettes avec Dreamweaver \(Macromedia\), j'obtiens des messages d'erreur quand je les installe sur le serveur.](#)
- [Le serveur qui m'héberge tourne sous PHP4, et ne comprend pas les fichiers « .php3 »](#)
- [J'obtiens des messages du genre « Maximum execution time exceeded »](#)
- [Dès ma première connexion à l'espace privé de SPIP, rien ne fonctionne et on m'indique une erreur à la ligne 1 de certains fichiers.](#)
- [SPIP ne fonctionne pas chez mon hébergeur habituel. Y a-t-il un « patch » quelque part ?](#)

• Le développement de SPIP et ses outils

- [Étendre SPIP](#)
- [Participer aux traductions](#)
- [Contribuer au développement de SPIP](#)

• Traduire SPIP

- [Comment traduire SPIP ?](#)
- [Traduire l'aide en ligne](#)
- [Conseils généraux pour la traduction de ce site](#)
- [Traduction des images](#)
- [À propos des forums](#)

FAQ Premiers pas
&
FAQ technique

FAQ Premiers pas

*****Lors de l'installation, la vérification des droits d'accès échoue à cause des répertoires CACHE et IMG. Pourtant, je suis sûr d'avoir réglé les droits correctement.*****

Il s'agit probablement d'un problème de majuscules. Vérifiez la façon dont ces deux répertoires ont été créés chez votre hébergeur. S'ils sont en minuscules, renommez-les en majuscules (avec votre logiciel de transfert de fichiers).

*****Mon site sous SPIP est planté, est-ce que je vous fais un procès ?*****

Nous diffusons SPIP gratuitement. Pour cette raison, nous insistons sur le point suivant : nous n'offrons aucune garantie d'aucune sorte, ni service après-vente (vu que SPIP n'est pas vendu...), ni garantie de résultat, ni obligation d'aide.

L'aide pour SPIP est donc basée sur le bénévolat et sur l'entraide des utilisateurs.

*****Je voudrais une nouvelle fonctionnalité...*****

Depuis [SPIP 1.9](#), il est possible d'intégrer des « [plugins](#) », qui rajoutent des fonctionnalités à SPIP.

Toutefois, si vous ne trouvez pas votre bonheur, vous pouvez écrire à la [liste des développeurs](#) pour la proposer, en donnant moult détails et en précisant les raisons qui vous poussent à vouloir une telle fonctionnalité.

N'oubliez pas cependant que SPIP est un logiciel libre, et que vous pouvez implémenter cette fonctionnalité vous-même si vous avez les compétences requises — auquel cas vous feriez des heureux en nous la communiquant !

*****Est-ce que je peux vendre un service utilisant SPIP ? Est-ce que je peux monter une activité professionnelle autour de SPIP ?*****

Vous faites bien ce que vous voulez...

Plus sérieusement, plusieurs utilisateurs nous ont déjà demandé si « ça ne nous embêtait pas » qu'ils utilisent SPIP dans le cadre de leur activité professionnelle (vendre des sites réalisés sous SPIP, vendre des formations à SPIP...). La réponse est claire : « ça ne nous embête pas », bien au contraire. Si nous avons choisi de diffuser SPIP sous GPL, c'est bien pour cela. Si SPIP peut vous aider économiquement, nous en serons plus que flattés.

*****Est-ce que je peux utiliser SPIP pour réaliser un site marchand, un site avec des pubs, ou un site de droite, ou un site qui débîne le logiciel libre...*****

Vous faites bien ce que vous voulez...

*****Comment réinstaller SPIP si je n'arrive plus à accéder à l'espace privé ?*****

Avec un client FTP (ou par tout autre moyen d'accès à votre espace d'hébergement), allez dans le dossier config/ récupérez sur votre poste le fichier connect.php puis effacez-le ou changez son nom. S'il existe, effacez aussi le fichier .htaccess. Puis suivez la procédure de configuration standard en visitant l'adresse <http://monsite.org/crire/> : « [Installer SPIP](#) ». Pour la connexion à la base de données, utilisez au besoin l'identifiant et le mot de passe contenus dans l'ancien fichier connect.php.

Historique : Jusqu'à [SPIP 1.9.1](#), le fichier à effacer s'appelait inc-connect.php3 (ou inc-connect.php) et se trouvait dans le dossier écrire/.

Important : cette procédure de réinstallation ne détruit pas la base de données (articles, rubriques, etc.). Elle sert simplement à recréer un accès administrateur au site, et à générer des fichiers d'authentification corrects.

*****De quoi a besoin SPIP pour fonctionner ?*****

SPIP a besoin (outre, bien sûr, un compte Web) de PHP3 ou 4, et d'une base de données MySQL. Voir la page « [Installer SPIP](#) » pour plus de détails.

Est-ce que cela marche sous tous les hébergeurs proposant les logiciels ci-dessus ? Quels sont les hébergeurs testés ?

Cela ne marchera pas forcément sous tous les hébergeurs car certains offrent une configuration extrêmement bridée (par exemple, obligation de créer les tables MySQL à la main).

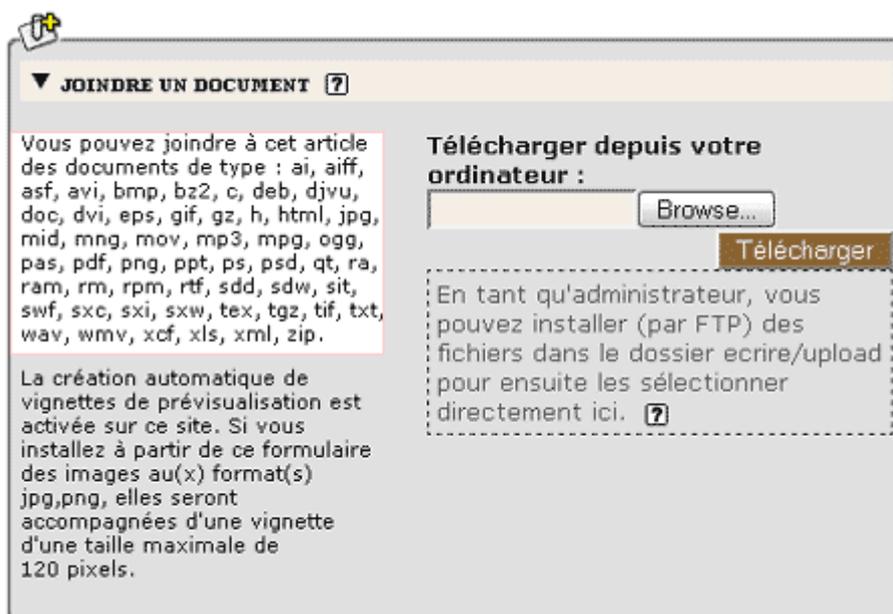
Nous établissons progressivement une [liste des hébergeurs acceptant SPIP](#). Consultez cette page pour plus d'informations.

FAQ technique

Ajouter un type de document

Depuis la version [SPIP 1.4], il est possible d'installer des documents joints aux articles (et, en option, aux rubriques).

Pour des raisons de sécurité, SPIP n'autorise pas l'installation de n'importe quels types de documents. En effet, permettre l'installation de documents sur un serveur distant à partir d'une interface Web peut poser de sérieux problèmes de sécurité. C'est pourquoi cette liste d'autorisations existe, et pour cette même raison SPIP ne propose pas d'interface pour modifier cette liste.



Le pavé permettant de joindre un document

Ce pavé affiche systématiquement la liste complète des types de documents autorisés (ici dans l'encadré clair).

Comme vous pouvez le constater, la liste des types de fichiers autorisés est déjà relativement fournie, et nous l'enrichissons régulièrement lorsque le besoin est exprimé sur les listes.

Avant de poursuivre, prenez le temps de bien lire ce qui suit :

- Cette manipulation est *potentiellement dangereuse* et peut introduire un énorme trou de sécurité dans votre site. Certains formats de fichiers (exécutables sur le serveur) ne doivent surtout pas être acceptés. En particulier, **n'acceptez jamais l'installation de fichiers de type PHP** (.php, .php3...) sur votre site, la sécurité de votre site serait *totaletement compromise*. N'hésitez pas à vous renseigner avant d'ajouter des types de fichiers.

- Volontairement, nous n'avons pas installé dans SPIP d'interface pour modifier la liste des types de fichiers autorisés. Cela pour réserver cette modification aux utilisateurs confirmés. Pour effectuer la manipulation, vous devez utiliser un gestionnaire de base de données (par exemple : phpMyAdmin). Or s'attaquer directement à la base de données de SPIP « à la main » (sans utiliser l'interface et les automatismes de SPIP) est potentiellement destructeur

pour votre site. Si vous ne savez pas exactement ce que vous faites, ne le faites pas. Si vous n'avez pas l'habitude de phpMyAdmin, ne l'utilisez pas sur la base de SPIP. Dans tous les cas, effectuez une sauvegarde de votre site SPIP avant de procéder à des interventions manuelles dans la base.

- Accessoirement, l'ajout d'un type de fichier ne se justifie que si les visiteurs de votre site peuvent utiliser (« lire ») ce type de fichier. Utiliser un format de fichier que l'on ne peut lire qu'avec un logiciel ultra-spécialisé sur un site grand public n'a pas vraiment de sens. Avant d'ajouter un type de fichier, vérifiez qu'il peut bien être utilisé par vos utilisateurs (notamment : lecteur ou plug-in gratuit et facile à installer ; ce format est-il lisible sur tous les ordinateurs, Mac, PC, Linux... ?).

- Pour toutes ces raisons, nous insistons sur le fait que **cette manipulation ne doit être effectuée qu'en parfaite connaissance de cause**. Si vous avez des doutes, renseignez-vous auprès de personnes compétentes ou, plus simplement encore, convertissez vos fichiers dans un format déjà autorisé et largement répandu (par exemple, un document FrameMaker ne pourra pas être lu directement par le grand public ; s'il s'agit simplement de diffuser son contenu, vous pouvez aussi bien en faire un fichier PDF directement téléchargeable par vos visiteurs plutôt que de vouloir ajouter le type « FrameMaker » dans la liste des fichiers autorisés par SPIP).

* *

Pour ajouter un type de fichier autorisé sur votre site sous SPIP, utilisez phpMyAdmin (ou tout logiciel équivalent) pour accéder à la gestion de la base de données.

Il vous faut ajouter un nouvel élément dans la table spip_types_documents.

←T→		id_type	titre	descriptif	extension	mime_type	Inclus	upload	maj
Edit	Delete	59	RealAudio		rm		embed	oui	20020829183952
Edit	Delete	58	RealAudio		ram		embed	oui	20020829183952
Edit	Delete	53	PDF		pdf		non	oui	20020829183952
Edit	Delete	3	GIF		gif		image	oui	20020829183952

Quelques types de documents dans phpMyAdmin

- id_type. Laissez ce champ vide. La numérotation des id_type est effectuée automatiquement par MySQL.

- titre. Indiquez ici le nom du type de fichier (souvent il s'agit du nom du programme qui permet de créer et lire ce type de fichier). Choisissez un nom court et aussi générique que possible. Sur un site multilingue, prenez soin à ne pas utiliser un titre propre à une langue (par exemple, un visiteur anglophone ne sera pas tellement intéressé par un fichier indiqué comme étant une « image vectorielle pour Illustrator » ; on se contentera donc d'indiquer « Adobe Illustrator »).

- descriptif. Laissez vide ; ce champ n'est pas utilisé.

- extension. Ce champ est le plus important : c'est là qu'on indique le type de fichier identifié par son extension. Par exemple, « rm » et « ram » pour du Real, « pdf » pour un fichier Acrobat PDF, « gif » pour une image au format GIF...

- mime_type. Laissez vide ; ce champ n'est pas utilisé.

- inclus. Le choix est laissé entre : « embed », « non », « image ». Prenez bien soin à effectuer ici le bon choix (c'est très important pour le bon fonctionnement de votre site) ; il détermine de quelle manière ce type de document sera « appelé » dans votre site pour être présenté aux visiteurs :

- « non » : ce type de fichier ne peut pas être inclus directement à l'intérieur d'une page

HTML ; on ne peut que le présenter au travers d'un lien hypertexte. Par exemple, un fichier PDF ne peut pas être affiché à l'intérieur d'une page HTML : le seul moyen de le joindre est de créer un lien hypertexte permettant d'ouvrir le fichier dans une nouvelle fenêtre (ou de le télécharger sur le disque dur du visiteur) ; un fichier de type « pdf » se voit donc attribuer le champ inclus fixé à « non » ;

- « embed » : ce type de fichier peut être directement affiché à l'intérieur d'une page HTML où il sera lu grâce à une extension du butineur (plug-in...). C'est le cas de la majorité des formats multimédia utilisés sur le Web : Flash, Shockwave, films vidéo...
 - « image » : il s'agit de formats d'image affichés directement dans la page HTML sans extension particulière (avec le code HTML). A priori, vous n'aurez pas besoin d'ajouter de tels types de fichiers, la liste fournie par SPIP étant déjà exhaustive. (Notez bien : certains formats d'images réalisés avec des logiciels de « dessin » ne peuvent pas être affichés directement en tant que « image » et nécessitent une extension pour être lus ; de tels formats seront alors « embed », voire « non ».)
- upload. Indique que vous autorisez l'installation de ce type de fichier via l'interface Web de SPIP. On choisit donc « oui ».
 - maj. Ce champ est géré automatiquement par SPIP. Laissez vide.

* *

Cette opération effectuée, vous pouvez créer une nouvelle icône (vignette de prévisualisation) correspondant à ce type de fichier. Pour assurer la cohérence graphique avec les icônes livrées par défaut avec SPIP, cette vignette aura une taille d'environ 48 pixels de large et 52 pixels de haut.

Vous pouvez utiliser tout type de format (GIF, PNG, JPG) ; préférez un format autorisant un fond transparent.

- Le nom de votre fichier sera formé ainsi :
 - l'extension du type de fichier autorisé ;
 - le format graphique de cette vignette (« .gif », « .png »...) ;
 - par exemple, une vignette sauvegardée en PNG, créée pour le format PDF, sera nommée : « pdf.png ».
 - N.B. Les noms se terminant par « -dist » sont réservés aux fichiers distribués avec SPIP. Ainsi, n'utilisez pas le nom « pdf-dist.png », ce nom est réservé au fichier créé par les développeurs de SPIP ; si vous utilisez tout de même ce nom, votre fichier risque d'être écrasé lors de votre prochaine mise-à-jour de SPIP.

Cette vignette s'installe par FTP dans le dossier /IMG/icones.

*****Des liens qui ouvrent une nouvelle fenêtre*****

Beaucoup d'internautes et de webmestres (surtout chez les débutants) estiment que certains liens sur les sites doivent ouvrir, de force, une nouvelle fenêtre dans le navigateur. L'équipe de SPIP, comme beaucoup d'autres, pense que c'est *mal*.

Toutefois, pour tenter de faciliter la vie à ceux qui voudraient absolument cette fonctionnalité, il existe un filtre [liens_ouvrants qui fait « sauter » tous les liens externes entrés dans le texte via le raccourci [titre-><http://www.uzine.net/spip>].

Au cours du développement de SPIP, aux alentours de septembre 2002, nous avons testé l'idée d'un raccourci de lien supplémentaire, idée que nous avons abandonnée après mûre réflexion : on n'y reviendra pas ;-)

Bilan des courses : dans les cas où c'est absolument nécessaire, il faut entrer le lien à la main, sans passer par le raccourci spip, c'est-à-dire en tapant `lien`.

Les autres méthodes sont à proscrire, voici pourquoi :

1) Un hack des raccourcis spip, qui ressemble à quelque chose du genre [texte->URL" target=...] est souvent signalé sur les listes spip comme une « solution ». Or c'est *la pire** des solutions. En effet, si jamais un jour on trouve quelque chose à faire des raccourcis de liens (par exemple, au pif, intégrer les liens dans une table et les vérifier périodiquement), ceux qui auront utilisé ce hack risquent de subir des dysfonctionnements.

2) Le raccourci [texte->>lien], qui a été testé entre le 21/09 et le 13/12 2002, dans les versions *de développement*, n'était pas satisfaisant. Il introduisait trop de problèmes de contrôle pour les webmestres qui n'en veulent pas.

La bonne solution, répétons-le, est de ne pas utiliser de liens ouvrants ; l'autre possibilité, moins bonne, est de les utiliser systématiquement sur les liens externes, en utilisant pour cela le filtre |liens_ouvrants.

*****Je suis chez Free ou chez Online, et j'obtiens souvent des messages « Maximum execution time exceeded »*****

Pour éviter qu'une erreur de programmation dans les scripts PHP ne bloque le fonctionnement du serveur, les hébergeurs fixent une durée maximale d'exécution de chaque page PHP. Cette limite est généralement fixée à 30 secondes : si une page « part » dans un calcul qui prend plus de 30 secondes (un énorme calcul, ou une boucle sans fin), le serveur interrompt le calcul et l'indique par un message d'erreur. Pour SPIP, cette limite de 30 secondes est largement suffisante pour effectuer tous ses calculs.

Certains hébergeurs fixent cependant des durées maximales très courtes. C'est notamment le cas chez Free (15 secondes) et chez Online (8 secondes). De plus, chez ces hébergeurs, la mémoire allouée au fonctionnement de PHP est plutôt faible, ce qui ralentit l'exécution. Ainsi, les fonctions de SPIP, chez ces hébergeurs, prennent plus de temps à s'exécuter, et la durée maximale pour qu'elles s'exécutent est ridiculement courte. Imaginez un train qui ne peut fonctionner que pendant une heure, et qui doit parcourir une distance de 120 kilomètres sur des lignes où la vitesse est limitée à 60 km/h : à mi-chemin le train s'arrêtera en rase campagne.

Ce problème se rencontre systématiquement avec des articles longs : chez ces hébergeurs, le calcul de telles pages est interrompu par le serveur avant la fin des calculs, ce qui interdit leur affichage et provoque le message d'erreur « Maximum execution time exceeded ».

Il n'y a aucune solution technique. SPIP a déjà été largement optimisé pour tourner plus vite, le problème est désormais du côté des hébergeurs.

- Seule solution immédiate : faites-vous héberger ailleurs, chez des hébergeurs qui fournissent un service utilisable, et non pas bridé d'une manière extrême.
- Ecrivez au responsable de l'hébergement pour lui demander d'augmenter la durée maximale d'exécution des scripts (30 secondes comme tout le monde, par exemple). On ne sait jamais, peut-être qu'à la longue... Mais pour l'instant, l'absence de réactions semble plus indiquer une volonté délibérée de brider l'hébergement gratuit pour pousser les webmestres à passer à l'hébergement payant qu'un réel impératif technique.

*****Pourquoi la création automatique de vignettes ne fonctionne pas avec tous les formats d'images (JPEG, GIF, PNG) ?*****

[SPIP 1.4] La création automatique de vignettes de prévisualisation utilise une fonctionnalité associée à PHP, nommée « GD » (on appelle cela une « librairie »). Cette librairie ne fait pas partie des fonctions intégrées à PHP, c'est une option [développée à part](#) (par Thomas Boutell), que les responsables des serveurs peuvent ou non installer avec PHP.

(1) Première conséquence : certains hébergeurs n'installent pas GD avec PHP. Chez ces hébergeurs, les fonctions graphiques sont donc inutilisables. Notamment SPIP ne pourra pas créer automatiquement les vignettes.

Dans ce cas, il faut demander à l'hébergeur d'installer la librairie GD.

(2) Le format GIF est un format propriétaire, qui utilise des fonctions qui ne sont pas des logiciels libres (notamment, leur propriétaire réclame une redevance aux programmes graphiques qui manipulent le format GIF). On ne peut donc pas l'installer dans un logiciel libre. Cependant, ignorant ce problème de copyright, les toutes premières versions de GD manipulaient le format GIF ; les versions récentes ne l'exploitent plus. Chez certains hébergeurs, avec des versions anciennes de GD, le GIF est donc utilisable ; mais chez la plupart, il n'est pas possible de l'exploiter.

Si le GIF n'est pas exploitable, il semble inutile de le réclamer à votre hébergeur : il faudrait en effet qu'il installe une ancienne version de GD plutôt qu'une version récente (autant demander à remplacer Windows 2000 par Windows 95 sur une machine de bureau).

(3) Pour fonctionner, la librairie GD utilise elle-même d'autres logiciels et bibliothèques pour traiter les formats PNG et JPEG. Si l'hébergeur n'a pas installé auparavant ces logiciels, GD fonctionnera, mais n'exploitera pas l'un ou l'autre de ces formats. Ainsi, même avec une version récente de GD, il est fréquent que seul le format PNG soit exploité, parce que les logiciels permettant de travailler en JPEG ne sont pas installés sur la machine...

Dans ce cas, demandez à votre hébergeur de compléter son installation pour que le format JPEG soit utilisable.

Dans la configuration précise de SPIP, lorsque vous demandez l'activation de la création automatique des vignettes, SPIP va tester les trois formats d'images, pour déterminer lesquels peuvent être exploités chez votre hébergeur. Si vous avez compris l'explication qui précède, vous savez que vous n'obtenez jamais l'utilisation des trois formats d'image ; fréquemment vous n'obtiendrez que le PNG et le JPEG ; parfois seulement le format PNG ; et le GIF encore plus rarement.

*****Comment fonctionne le moteur de recherche de SPIP ?*****

Le moteur de recherche intégré à SPIP est très simple d'utilisation, et cependant relativement puissant. Même si la plupart des utilisateurs n'ont aucune raison de se demander « comment ça marche ? », nombreux sont les courriers qui demandent des précisions sur son fonctionnement...

Voici les principes sur lesquels repose le moteur de SPIP.

Afin d'être rapide et efficace (c'est-à-dire pour qu'il donne des réponses pertinentes), le moteur de SPIP utilise un système d'**indexation** des contenus. L'indexation consiste à analyser tous les textes contenus dans SPIP, à en extraire tous les mots, et à sauvegarder pour chaque mot l'endroit où il apparaît.

Comme l'index dans un livre vous présente les mots importants du livre, suivis des numéros des pages où les retrouver, l'index du moteur de recherche fait correspondre à chaque mot utilisé sur le site le numéro de l'article, de la brève... où le retrouver.

Ensuite, lorsqu'on utilise le moteur pour effectuer une recherche, SPIP n'a plus besoin de consulter l'ensemble des textes du site, mais simplement d'aller consulter l'index pour savoir où se trouvent ces mots. On gagne ainsi énormément de temps (et plus le site est gros, plus on gagne de temps).

L'indexation

Le principe est donc le suivant : prendre un texte (plus ou moins long), en extraire tous les mots, et noter chacun de ces mots dans une base de données, en faisant correspondre ce mot à l'endroit où il se trouve.

Par exemple, notre site a trois articles, dont les textes (très courts) sont :

- *article 1* : « Le petit chat est mort de froid et de faim. »
- *article 2* : « Le gros chat est rentré à la maison. »
- *article 3* : « La maison résiste au froid. »

Nous allons extraire les mots de chaque article, et enregistrer pour chaque mot à quel article il correspond (nous ne prendrons que les mots de plus de trois lettres, nous expliquerons plus loin pourquoi) :

- *petit* : 1
- *chat* : 1, 2
- *mort* : 1
- *froid* : 1, 3
- *faim* : 1
- *gros* : 1
- *rentré* : 2
- *maison* : 2, 3
- *résiste* : 3

Et ainsi de suite, en considérant que notre site est certainement beaucoup plus gros, et les articles beaucoup plus long.

Si l'on recherche le mot *chat* :

- une solution sans indexation consisterait à relire tous les articles, pour y trouver le mot *chat* ; sur un gros site, même pour un ordinateur, cela prend beaucoup de temps ;
- puisque nous avons un index, il suffit de consulter l'entrée *chat* : on sait immédiatement qu'il se trouve dans les articles 1 et 2.

La pondération

À l'indexation s'ajoute un deuxième principe : la pondération. Il s'agit d'essayer de rendre le moteur plus pertinent. Par exemple, si un mot apparaît dans le titre d'un article, et dans le corps du texte d'un autre article, on considère que si l'on recherche ce mot, il faut en premier indiquer celui où il apparaît dans le titre. De plus, si un mot apparaît 25 fois dans un article, et seulement deux fois dans un autre, on veut afficher en premier l'article où le mot est le plus fréquent.

On voit que l'indexation simple ne suffit pas. Si on recherche *chat*, on trouvera les articles où il apparaît, mais sans pouvoir ensuite classer ces articles entre eux (selon que le mot chat apparaît une fois ou 20 fois, ou s'il se trouve dans le titre ou seulement dans le texte...).

SPIP va donc calculer une pondération pour chaque mot dans chaque article. C'est-à-dire donner des points à ce mot en fonction de l'endroit où il se trouve, et du nombre de fois où il apparaît :

dans le titre	8 points
dans le soustitre	5 points
dans le surtitre	5 points
dans le descriptif	4 points
dans le chapo	3 points
dans le texte	1 point
dans le post-scriptum	1 point

Si le mot apparaît plusieurs fois, on additionne les occurrences.

Par exemple, si dans un article, le mot *chat* apparaît :

- une fois dans le titre : 8 points
- deux fois dans le descriptif : 2 fois 4 = 8 points
- six fois dans le texte : 6 fois 1 = 6 points
- *total* : 8 + 8 + 6 = 22 points.

Le mot *chat*, dans l'index, est donc ainsi enregistré :

- *chat*, dans l'article numéro X, 22 points ;
- *chat*, dans l'article numéro Y, 15 points ;
- ...

Si l'on recherche le mot *chat*, grâce à l'index, on saura donc qu'il se trouve dans les articles X et Y, et de plus on peut classer ces articles entre eux : 22 points dans X, 15 points dans Y, donc on considère que l'article X répond mieux à la recherche.

Les mots-clés : beaucoup d'utilisateurs confondent les mots-clés et l'indexation. Les mots-clés n'ont, [par nature](#), aucun rapport avec l'indexation : quand SPIP effectue une recherche, il ne recherche pas dans les mots-clés associés à des articles (ce serait très limité), il recherche dans l'index qu'il a fabriqué à partir du texte exact des articles. Donc, dans le principe même de l'indexation, les mots-clés n'interviennent pas du tout.

En revanche, les mots-clés sont tout de même utilisés pour fabriquer la pondération des articles. Si un mot-clé est associé à un article, il entre alors dans l'indexation de l'article lui-même, avec une forte pondération (12 points pour le nom du mot-clé, 3 points pour son descriptif). Ainsi, si notre article Y (15 points en prenant simplement compte son contenu propre) est associé au mot-clé *chat* (parce qu'on indique par ce mot que c'est le sujet de cet article), il faut ajouter à l'indexation de cet article, pour le mot *chat*, les 12 points du mot-clé : total 27 points. Dans notre recherche sur *chat*, l'article Y devance désormais l'article X.

Notons enfin que tous les éléments de contenu de SPIP font ainsi l'objet d'une indexation : les articles, les brèves, les rubriques, les auteurs, les mots-clés, les sites référencés (si le site est une syndication, les titres des articles récupérés sur ce site entrent également dans l'indexation).

Où est-ce stocké ?

Les données de l'indexation sont directement stockées dans la base de données. Cette partie est un peu technique, aussi je ne m'étendrai pas longtemps sur le sujet.

Sachez simplement qu'il y a plusieurs index (listes des mots), correspondant chacun à un type de contenu (un index pour les articles, un index pour les rubriques, un index pour les brèves...). De plus, contrairement à l'explication ci-dessus, où les entrées de l'index sont des mots, dans SPIP les entrées des index sont des nombres calculés à partir de ces mots (des *hachages*) ; une autre table de la base de données stockant par ailleurs les correspondances entre les véritables mots et ces nombres ; cette méthode permet d'accélérer les recherches dans les index (il est plus rapide pour le logiciel de rechercher des nombres plutôt que des mots).

Notez surtout que la taille des index est très importante. Sur uZine, par exemple, la partie de la base de données consacrée au stockage des articles pèse 9,7 Mo. La partie qui stocke l'index correspond aux

articles pèse 10,5 Mo. Et la partie qui stocke la correspondance entre les mots et leur traduction en nombres 4,1 Mo. Ainsi, pour un site dont les articles occupent 9,7 Mo, l'indexation de ces articles représente, à elle seule, près de 14,6 Mo. L'espace nécessaire à ces articles et à la recherche a donc plus que doublé la taille occupée par le site. C'est l'une des raisons, notamment, pour lesquelles on peut préférer désactiver le moteur de recherche, si l'on a d'importantes limitations en matière d'hébergement.

Quels mots sont indexés ?

Nous l'avons vu, tous les mots de tous les éléments de contenu du site sont extraits, puis analysés (pour pondération). Cependant, SPIP ne va pas retenir tous les mots.

- Les codes HTML sont exclus de l'indexation. Cela est évidemment nécessaire pour obtenir des recherches « propres »...
- Les mots de moins de 4 lettres ne sont pas retenus (en réalité, de moins de 3 lettres, mais les mots de 3 lettres ne sont pour l'instant pas exploités réellement lors des recherches). Ce point soulève beaucoup de questions de la part des utilisateurs...

Le problème est d'obtenir des résultats aussi pertinents que possible. Il faut donc privilégier, dans les recherches, les mots réellement importants dans le sens de la recherche. Par exemple, si l'on recherche les mots *le chat*, le mot important est *chat*, et non *le*...

Reprenons notre premier exemple (avec trois articles constitués chacun d'un phrase). Si l'on avait indexé tous les mots, nous aurions notamment les mots :

- *le* : 1, 2
- *est* : 1, 2
- ...

En recherchant les mot *le froid est dangereux*, nous trouverions les entrées :

- *le* : 1, 2
- *froid* : 1,3
- *est* : 1, 2
- *dangereux* : nulle part.

En ajoutant les résultats de ces mots pour chaque article (en réalité, la pondération de chaque article, mais considérons pour notre exemple que chaque mot a une pondération d'un seul point), on obtiendrait :

- article 1 : 3 mots
- article 2 : 2 mots
- article 3 : 1 mot.

Le classement mettrait donc en tête l'article 1, puis l'article 2, puis l'article 3. Or, l'article 2 ne parle pas de froid, contrairement à l'article 3. À cause de l'utilisation des mots sans importance pour le sens (*le*, *est*), le classement est faussé.

D'où le besoin, dans l'indexation, de ne pas tenir compte des mots qui n'entrent pas dans le sens de la recherche. La méthode la plus propre consiste à fournir au système une liste de mots à ne pas indexer ; cependant, cela nécessite un travail énorme, c'est-à-dire la constitution de dictionnaires d'exclusion (et cela dans toutes les langues)... Donc, plus simplement, dans SPIP nous choisissons de considérer que les mots de trois lettres et moins ne sont pas « importants » ; évidemment, il y a de la casse, puisque des mots comme *Val*, *mer*, *sud*... ne sont plus pris en compte ; c'est donc un compromis, qui se juge sur l'efficacité des recherches (qui sont globalement de bonne qualité).

N.B. Depuis la version 1.6 de SPIP, les sigles de deux lettres et plus, y compris ceux contenant des chiffres (G8, CNT, PHP, ONU...), sont indexés. Un sigle est un mot ne comprenant aucune minuscule.

Quand a lieu l'indexation ?

L'indexation a lieu à trois moments différents :

- lorsque vous validez un article, celui-ci est immédiatement indexé ;
- lorsque vous modifiez un article déjà publié, il est à nouveau indexé ;

sauvegarde dans une nouvelle version de SPIP. Il ne faut surtout pas installer une vieille sauvegarde dans une nouvelle version de SPIP. Il ne faut surtout pas installer une vieille sauvegarde dans une nouvelle version de SPIP. Il ne faut surtout pas installer une vieille sauvegarde dans une nouvelle version de SPIP. »

Allez savoir pourquoi, un jour un individu a annoncé sur la liste de SPIP qu'il fallait faire comme cela et, malgré toutes nos explications contraires, nombre d'utilisateurs restent persuadés qu'il faut réinstaller sa vieille sauvegarde quand on change de version de SPIP.

C'est une erreur *fatale*, pire qu'un virus dans votre Outlook. Votre ordinateur prend feu, le disque dur fond, votre hébergeur fait faillite, France Telecom se trompe dans votre facture, et vous êtes interdits à vie de jamais plus utiliser le net.

La procédure correcte pour effectuer une mise à jour, [c'est expliqué là](#).

*****Je viens d'effectuer une mise à jour de SPIP, et le site public fonctionne mal*****

Il y a trois raisons principales pour qu'une mise à jour provoque des messages d'erreur.

1. L'espace privé fonctionne parfaitement, sans erreurs. Seul le site public déconne. C'est normal... En effet, certaines « fonctions » de SPIP ont été modifiées (sinon il n'y aurait pas mise à jour) ; dans le même temps, les pages du site public sont stockées pendant un certain temps en cache. Au moment de la mise à jour, il peut arriver que les pages du site public (calculées avant la mise à jour) utilisent des fonctions dont le fonctionnement a changé.

La solution la plus simple : **vider le cache**. Rendez vous sur la page « Sauvegarde/restauration de la base », et cliquez sur le bouton « Vider le cache ».

2. Vous avez récupéré la dernière version beta-x24 planquée au fin fond de notre serveur, pour voir à quoi elle ressemble.

Si vous n'avez aucune connaissance en PHP/mysql, c'est une très mauvaise idée. Les version beta sont des versions en cours de développement, non terminée. Certains jours et avec beaucoup de chance, elles sont relativement stables ; d'autres jours, elles contiennent des bouts de fonctions pas terminées, et surtout, pas déboguées.

Deux cas de figure :

- vous avez effectué une sauvegarde de votre base de données (page « Sauvegarde / Restauration de la base ») avant la mise à jour malencontreuse ; dans ce cas, toujours sur cette page, effacez totalement la base de données (bouton « Effacer tout »). Réinstallez la version précédente de SPIP (important : la version avec laquelle vous avez effectué la sauvegarde), et réinstallez le site à partir de votre sauvegarde ;
- vous n'avez pas de sauvegarde de votre base de données... grave erreur. Si vous n'êtes pas bricoleur, le mieux à faire est encore d'attendre quelques jours la prochaine version beta, en espérant qu'elle soit plus stable (généralement, les versions beta ne restent pas totalement instables très longtemps ; il y a toujours un moment où, grosso modo, elles fonctionnent : mais souvent l'interface n'est pas terrible...).

3. Après la mise à jour de SPIP, vous avez voulu réinstaller la sauvegarde de votre base effectuée avant la mise à jour. On se demande pourquoi tant de monde a cette idée saugrenue...

Nous consacrons [une page spécifique](#) expliquant comment « bien » effectuer une mise à jour.

Lorsque SPIP change de version, il est fréquent que, pour installer les nouvelles fonctionnalités et améliorer le fonctionnement du système, nous modifions la structure de la base de données, c'est-à-dire la

façon dont les informations de votre site sont stockées. Vouloir réinstaller une ancienne sauvegarde dans une nouvelle version de SPIP, c'est un peu comme faire passer des carrés dans des trous triangulaires.

La seule solution « propre » pour rattraper cela : rendez-vous dans la page « Sauvegarde / restauration de la base », et effacez la base de données (« Effacer tout »). Installez l'ancienne version de SPIP (toutes les versions sont accessibles dans les archives), restaurez la base à partir de votre sauvegarde. Cela fait, vous pourrez installer la nouvelle version de SPIP, le système effectuera de lui-même les modifications nécessaires sur la structure de votre site.

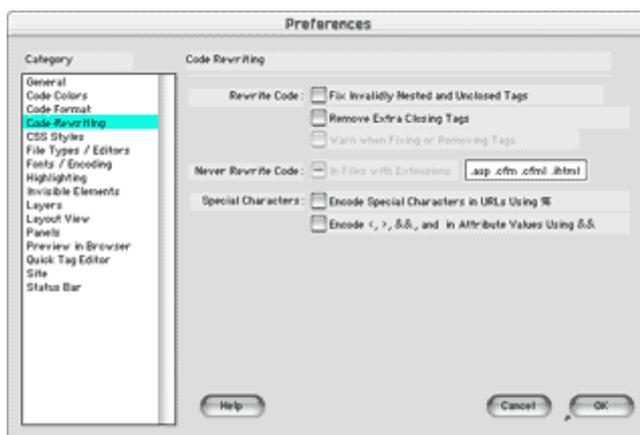
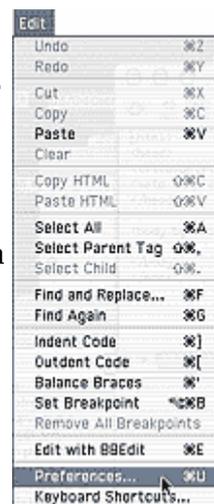
*****Après modification de mes squelettes avec Dreamweaver (Macromedia), j'obtiens des messages d'erreur quand je les installe sur le serveur.*****

De nombreux utilisateurs de SPIP utilisent Dreamweaver (de Macromedia) pour modifier les *squelettes* de mise en page de leur site. Ce choix peut en effet se justifier par le fait que Dreamweaver permet à la fois de travailler en mode graphique (WYSIWYG) et d'accéder immédiatement au code source (ce qui est indispensable pour travailler proprement sur les squelettes).

Cependant, Dreamweaver a une fâcheuse tendance à *modifier* le code des squelettes et à perturber le fonctionnement des boucles. Les codes spécifiques à SPIP ne sont en effet par reconnus comme du code HTML, et Dreamweaver essaie de les corriger, ce qui interdit ensuite le bon fonctionnement de vos pages (on obtient un message d'erreur : « tag fermant manquant »).

La solution consiste tout simplement à interdire à Dreamweaver de corriger « tout seul » le code source :

- dans le menu « Edit », sélectionnez « Preferences » ;
- sélectionnez l'onglet « Code rewriting », et décochez toutes les options.



Cela fait, Dreamweaver ne devrait plus modifier votre code source à votre insu.

Et voici un autre « truc » pour que l'affichage des codes spip soit correct :

Coller dans/dreamweaver/configuration/ThirdPartyTags

- deux images nommées SPIP.gif (par exemple redacteurs-off.gif et redacteurs-on.gif)
- un fichier spip.xml contenant :

```

<tagspec tag_name="Boucle_1" start_string="<BOUCLE" end_string=">"
detect_in_attribute="true" icon="SPIP.gif" icon_width="17"
icon_height="15"></tagspec>

<tagspec tag_name="Boucle_0" start_string="</BOUCLE_" end_string=">"
detect_in_attribute="true" icon="SPIP.gif" icon_width="17"
icon_height="15"></tagspec>

<tagspec tag_name="B_0" start_string="<B_" end_string=">"
detect_in_attribute="true" icon="SPIP2.gif" icon_width="17"
icon_height="15"></tagspec>

<tagspec tag_name="B_1" start_string="</B_" end_string=">"
detect_in_attribute="true" icon="SPIP2.gif" icon_width="17"
icon_height="15"></tagspec>

<tagspec tag_name="B_2" start_string="</B_" end_string=">"
detect_in_attribute="true" icon="SPIP2.gif" icon_width="17"
icon_height="15"></tagspec>

```

Les balises SPIP seront alors remplacées par l'image, et éditables dans la fenêtre « propriété de l'objet ».

P.-S.

A titre d'exemple, voici un paquet qu'il vous suffit de dézipper dans le répertoire /dreamweaver/configuration/ThirdPartyTags pour obtenir le résultat décrit ci-dessus.



Fichiers pour Dreamweaver

Dézipper les fichiers de l'archive dans le répertoire "ThirdPartyTags" de Dreamweaver, et c'est tout !

*****Le serveur qui m'héberge tourne sous PHP4, et ne comprend pas les fichiers « .php3 »*****

Dans la plupart des cas, les serveurs utilisant PHP4 sont configurés pour reconnaître les fichiers utilisant la terminaison « .php3 », afin d'assurer la compatibilité. Si tel n'était pas le cas (vous voyez apparaître le

texte source à l'écran, ou bien votre butineur vous dit qu'il ne reconnaît pas le format du fichier), contactez votre hébergeur pour qu'il effectue la manipulation.

[Karl](#) signale, à l'attention de ceux qui contrôlent leur propre serveur : Si vous avez accès à votre httpd.conf et que vous avez compilé PHP4 modifiez ainsi votre httpd.conf :

```
## And for PHP 4.x, use : #  
AddType application/x-httpd-php .php .php3  
AddType application/x-httpd-php-source .phps
```

*****J'obtiens des messages du genre « Maximum execution time exceeded »*****

Cela signifie que le temps de calcul d'une page (soit sur le site public, soit sur la partie privée) a dépassé la limite fixée par votre hébergeur.

Cette limite est souvent fixée à 30 secondes, ce qui est normalement suffisant pour SPIP, et ce message ne devrait pas apparaître. En revanche, certains hébergeurs (généralement les gratuits) fixent une limite très faible. Par exemple, sur Free.fr, cette limite est de 5 secondes. Sur le calcul de pages longues ou compliquées sur un serveur plutôt lent, vous pouvez donc voir apparaître cette erreur.

Si vous avez accès à la configuration de php (fichier /etc/php/apache/php.ini sur Linux), il faut modifier le paramètre *max_execution_time*.

Enfin, si le problème perdure, il vous reste à enquiquiner votre hébergeur pour qu'il fixe une limite qui permette de travailler sérieusement.

*****Dès ma première connexion à l'espace privé de SPIP, rien ne fonctionne et on m'indique une erreur à la ligne 1 de certains fichiers.*****

Lorsque vous avez transféré vos fichiers vers votre hébergeur, par FTP, vous avez sélectionné un mauvais format de fichier. Essayez en indiquant à votre client FTP un autre format (format binaire, texte...).

De temps en temps, c'est nous-même qui faisons l'erreur lors de l'installation de SPIP sur notre serveur public, et les fichiers ne sont pas utilisables ! Dans ce cas, nous postons rapidement une nouvelle version corrigée.

*****SPIP ne fonctionne pas chez mon hébergeur habituel. Y a-t-il un « patch » quelque part ?*****

Avant tout, vérifiez que le problème vient bien de votre hébergeur. Utilisez notamment la [liste de discussion de SPIP](#) pour demander à d'autres utilisateurs s'ils ont eu des difficultés similaires avec cet hébergeur.

Dans un second temps, si le problème que présente cet hébergeur est identifié, et dépend d'un choix de configuration de sa part, n'hésitez pas à contacter le responsable pour lui demander de modifier sa configuration pour qu'elle soit compatible avec SPIP ; certains responsables de serveurs d'hébergement

ont déjà fait des efforts pour aider leurs utilisateurs sous SPIP.

Finalement, en dernier recours, votre hébergeur ne veut ou ne peut pas modifier sa configuration pour qu'elle ne soit plus incompatible avec SPIP. Est-ce qu'il existe des « patches » spécifiques pour certains hébergeurs, c'est-à-dire des petits bouts de programmes, des modifications spécifiques dans SPIP ? *Non*. Le but de l'équipe de développement est d'éviter la multiplication des « patches » et des bidouilles dans le code, afin de livrer une unique version utilisable par tous (la facilité d'installation de SPIP est l'une de nos priorités). Si votre hébergeur présente un problème spécifique, nous verrons s'il y a moyen de le résoudre, et nous incorporerons la solution dans la version suivante de SPIP.

Le développement de SPIP et ses
outils

Étendre SPIP

Si vous voulez étendre SPIP et en particulier y contribuer, les points importants à retenir sont les suivants : *c'est un outil déjà utilisé par des milliers de personnes, qui a un passé et un avenir, et qui est un travail collectif*. Certains choix ont été opérés, pas toujours heureux, mais qui donnent une cohérence à l'ensemble du code, facilitant son évolution. Lorsqu'un choix se révèle franchement incompatible avec une nouvelle technologie intéressante à intégrer, une réécriture est mise en chantier mais peut prendre plusieurs mois, d'où une incohérence prolongée. On comprend donc qu'il faut éviter autant que possible de telles décisions, et accepter qu'ici où là une nouvelle règle soit violée, la réécriture de cette partie du code n'ayant pas encore eu lieu. On comprend aussi que le style de programmation varie quelque peu d'un endroit à un autre, mais que se construit une identité commune en perpétuel devenir, un logiciel utile et utilisé n'étant pas de saintes écritures figées sur papier bible, mais l'expression d'une communauté vivante et libre.

Organisation des sources

Depuis la version 1.9, il est possible de modifier le comportement de SPIP sans altérer ses sources, grâce au *chemin d'accès* défini par la constante `SPIP_PATH`. Toute extension de SPIP doit opérer ainsi afin qu'un test d'intégration soit débrayable par simple redéfinition du chemin d'accès, la distribution de SPIP étant supposée en lecture seule. Si vous êtes convaincu qu'un certain comportement ne peut être obtenu ainsi, écrire à la liste de discussion [spip-dev](#).

Les contributions à SPIP sont à décrire par un article sur [Spip-Contrib](#) et leur contenu est à déposer soit en tant que pièces jointes à cet article, soit, ce qui est mieux, sur [spip-zone](#) qui fournit un serveur [Subversion](#) piloté par [Trac](#).

Pour étendre SPIP ainsi, il faut bien comprendre l'organisation de ces répertoires et le rôle de ses fichiers. C'est le but du présent article, mais on lira aussi avec profit l'article annonçant SPIP 1.9 qui l'a inaugurée.

La racine d'une distribution de SPIP comporte essentiellement :

- un fichier `spip.php`, alias `index.php`, gérant la compatibilité avec les anciennes versions, chargeant le fichier d'initialisation `ecrire/inc_version.php` et passant immédiatement la main au script principal `ecrire/public.php` ;
- un répertoire `ecrire` comportant exclusivement des fichiers interprétables côté serveur (PHP et MySQL) ;
- un répertoire `dist` comportant exclusivement des fichiers interprétables côté client (HTML, Javascript, feuilles de style, images de différents formats) ainsi que les patrons de mise en page nommés *squelettes*. Ces squelettes sont interprétés des deux côtés : il s'agit de fichiers composés d'un format MIME complété de quelques directives SPIP, directives traitées côté serveur afin d'envoyer au client un texte purement MIME (la plupart du temps du HTML, mais aussi du RSS, du SVG, du ICS etc).

Selon la manière dont SPIP est installé, d'autres répertoires peuvent figurer à la racine mais sont vides à l'installation : ils contiendront les données, temporaires ou permanentes, nécessaires à la vie du site.

Rôles du répertoire dist et ses sous-répertoires

répertoire	rôle
/	contient les squelettes. Leur nom se termine par .html pour des raisons historiques mais cela ne préjuge pas de leur contenu. Il suffit de donner un tel nom, privé de cette extension, au paramètre page de l'URL d'un site sous SPIP pour déclencher l'utilisation de ce squelette. La racine du répertoire dist/ contient également les feuilles de styles (d'extension .css) définissant l'habillage par défaut
formulaires/	contient la partie html des <i>balises dynamiques</i> , squelettes de formulaires dont le code PHP figure dans le répertoire ecrire/balise
icones_barre/	contient les images destinées à illustrer la barre typographique
images/	images de l'espace privé
javascript/	librairies javascript (jQuery, barre typographique, ...)
modeles/	squelettes appelables avec la balise #MODELE ou avec les raccourcis <article6 modele>
polices/	polices utilisables pour les images typographiques
vignettes/	vignettes standards pour les types de documents pouvant être associés à un article

Rôles du répertoire ecrire et ses sous-répertoires

Le répertoire ecrire comporte plusieurs sous-répertoires composés de fichiers PHP définissant des fonctions et procédant éventuellement, mais rarement, à une initialisation lors de leur chargement (ces exceptions sont appelées à disparaître). Les fichiers situés au niveau du répertoire principal sont les plus importants à comprendre pour contribuer à SPIP, il s'agit de inc_version.php, public.php et index.php.

Le fichier ecrire/inc_version.php initialise les constantes et les variables globales nécessaires au fonctionnement de SPIP, notamment celles assurant sa portabilité sur les différentes plates-formes. Assez tôt lors de son chargement, il inclut le fichier inc/utls.php, où figurent les fonctions indispensables à SPIP, un fichier hors distribution nommé mes_options.php permettant de moduler cette initialisation sans avoir à modifier le fichier inc_version.php. En particulier, il est possible dans ce fichier personnel d'invoquer la fonction spip_initialisation pour définir les répertoires de données et disposer ainsi de plusieurs sites sous SPIP utilisant une seule distribution (l'appel standard de cette fonction, plus loin dans inc_version.php, sera automatiquement neutralisé). Une autre fonction indispensable à SPIP est find_in_path, exploitant le *chemin d'accès*, ainsi que include_spip qui repose sur find_in_path, et charger_fonction qui repose sur include_spip. Tous les fichiers de Spip sont chargés par ces deux dernières fonctions.

Le fichier ecrire/public.php, appelé par spip.php, a pour rôle essentiel de délivrer les pages de l'espace public, commandées lorsque la requête HTTP comporte (après réécriture éventuelle) le paramètre page. Ce script applique alors le squelette ayant pour nom la valeur de ce paramètre. Il envoie les en-têtes HTTP et le contenu obtenus, gère leurs éventuelles erreurs et lance les tâches de fond à l'aide de la fonction cron. Contribuer à l'espace public de SPIP consiste donc simplement à fournir de nouveaux squelettes, avec éventuellement leurs feuilles de style et leurs images.

L'autre rôle de ecrire/public.php concerne le cas où la requête HTTP comporte l'argument action. Il applique alors la fonction charger_fonction à la valeur v de ce paramètre exec. Cette application a pour effet de charger le fichier homonyme du répertoire action, dont la fonction principale action_v_dist est alors invoquée. Ces scripts effectuent des tâches diverses ne retournant en général pas de résultat, échappant ainsi à la problématique de la mise en page.

Le fichier index.php est le fichier central d'accès aux formulaires de l'espace privé. Il authentifie l'internaute, initialise ses données personnelles et applique la fonction charger_fonction à la valeur v du

paramètre `exec`. Cette application a pour effet de charger le fichier homonyme du répertoire `exec`, dont la fonction principale `exec_v_dist` est alors invoquée. Celle-ci a la charge de délivrer l'intégralité du flux de sortie, y compris les en-têtes HTTP. Il est donc possible d'étendre Spip simplement en rajoutant un fichier PHP dans un sous répertoire nommé `exec` d'un répertoire figurant dans `SPIP_PATH`.

Le répertoire `exec` contient exclusivement les fichiers définissant les fonctions directement invocables par le paramètre d'URL `exec`. Le code PHP de ces fichiers ne doit jamais accéder en écriture à la base de données (les exceptions à cette règle sont en voie de disparition). A l'inverse, il y accède abondamment en lecture afin de vérifier les droits du demandeur et déterminer les données à visualiser. Si l'on veut voir SPIP sous l'archétype *Modèle-Vue-Contrôleur*, les fichiers de `exec` remplissent le rôle de *Contrôleur*. Si l'on veut voir SPIP sous l'archétype (*Print(Eval(Read))*) de Lisp, c'est la partie *Read*. A terme, ce répertoire devrait devenir un répertoire de squelettes. Il est demandé aux nouvelles contributions de Spip de penser à cet objectif lors de leur rédactions.

Le répertoire `action`, dont il a déjà été question, contient essentiellement les scripts accédant en écriture à la base de données. Si l'on veut voir SPIP sous l'archétype *Modèle-Vue-Contrôleur*, les fichiers de `action` remplissent le rôle de *Modèle*. Si l'on veut voir SPIP sous l'archétype (*Print(Eval(Read))*) de Lisp, c'est la partie *Eval*. Là encore, contribuer à SPIP consiste à écrire de tels scripts et à les invoquer par des formulaires construits avec la fonction `generer_action_auteur` assurant la sécurisation de l'accès à ces scripts, qui à l'inverse invoqueront la fonction `controler_action_auteur` pour vérifier les droits du demandeur. Cette architecture permet de calculer ces droits seulement à la construction des formulaires appelant les scripts d'accès en écriture : plutôt que de recalculer tous les droits, ces scripts vérifieront simplement que la clé figurant dans les arguments est la même que celle qu'ils recalculent à partir des autres arguments, de l'identité du demandeur et d'une valeur aléatoire renouvelée périodiquement. Ces scripts ne retournent en général pas de résultat, il n'y a donc là aussi ni code HTML, ni appel à la fonction `echo` (les exceptions sont appelées à disparaître). En revanche, ils sont souvent appelés avec un paramètre HTTP nommé `redirect`, demandant une redirection qui sera alors automatiquement opérée par `public.php`, qui envoie un statut HTTP 204 en l'absence de ce paramètre. Dans le cas des formulaires construits avec la fonction `ajax_action_auteur`, cette redirection conduit au script homonyme dans le répertoire `exec`. Ce deuxième script se réduit le plus souvent à charger le fichier homonyme dans le répertoire `inc`, d'appeler sa fonction principale dont le résultat est retourné au client par l'intermédiaire de la fonction `ajax_retour`. Il est ainsi très facile d'étendre SPIP en mode AJAX en utilisant de cette infrastructure.

Le répertoire `inc`, le plus volumineux, contient essentiellement les fonctions construisant les pages de l'espace privé renvoyées au client, ces fonctions devant être à terme les filtres utilisés par les fichiers de `exec` quand ils seront des squelettes. Si l'on veut voir SPIP sous l'archétype *Modèle-Vue-Contrôleur*, les fichiers de `inc` remplissent le rôle de *Vue*. Si l'on veut voir SPIP sous l'archétype (*Print(Eval(Read))*) de Lisp, c'est la partie *Print*. Toutefois ce répertoire contient également beaucoup de fonctions relevant plutôt du *Contrôle* et devra donc être réorganisé. La plupart des fichiers de `inc` sont chargés par l'intermédiaire de `charger_fonction`, et ce sera le cas de tous à terme. Aucune des fonctions de ce répertoire n'est censée utiliser `echo`. Les contributions à SPIP sont appelées à respecter ces règles dès à présent.

Le répertoire `install` contient exclusivement les fonctions nécessaires à l'installation de SPIP. Chaque étape peut-être surchargée ou complétée par d'autres, la fonction principale de `exec/install.php` utilisant ce répertoire selon le même principe que `ecrire/index.php` avec le répertoire `exec`.

Le répertoire `urls` contient des fichiers définissant chacun le même jeu de *fonctions de réécriture d'URL*. Il s'agit des fonctions calculant, à partir d'un index numérique dans une table de la base de données, un signet plus facile à lire et écrire que l'appel du script PHP effectivement opéré par le serveur HTTP pour cet index et cette table. Là encore, il suffit de rajouter un fichier dans ce répertoire pour obtenir un nouveau jeu, dont le nom sera affecté à la globale `type_urls`.

Le répertoire `lang` contient exclusivement des fichiers de données, tableaux indiquant la traduction, pour toutes les langues connues de SPIP, de tous les arguments que la fonction `_T`, définie dans `inc/utls.php`, est susceptible de recevoir. Les fichiers sont chargés exclusivement par les fonctions de `inc/lang.php`. Traduire les fichiers de référence `*fr*` en donnant un nom conventionnel aux fichiers obtenus suffit à déclarer une nouvelle langue à SPIP.

Le répertoire charset contient lui aussi exclusivement des fichiers de données, tableaux permettant de passer d'un codage de caractères à un autre (utf, iso, ascii, entités html etc). Ils sont lus exclusivement par les fonctions de inc/charsets.php. Il suffit là encore de rajouter un fichier pour disposer d'un nouveau codage, mais SPIP propose tous ceux couramment utilisés, aussi une telle intervention est rarissime.

Le répertoire base contient les fonctions d'interfaces entre PHP et la base MySQL installée. Les contributions possibles sont essentiellement le portage vers d'autres serveurs de base de données. SPIP n'a pas été prévu pour cela au départ, aussi l'interface pour de telles contributions est encore lacunaire. On regardera néanmoins, malgré son incomplétude, le fichier générique abstract_sql.php, et son application à MySQL dans db_mysql.php si l'on veut tenter l'aventure (des portages partiels ont été réalisés ainsi). Ne doit évidemment figurer aucun code MIME dans ce répertoire.

Le répertoire balise contient les fichiers PHP associés aux *balises dynamiques* de SPIP. Leur nom est homonyme du squelette de dist/formulaires. Compléter l'espace public de SPIP par un formulaire *F* consiste à créer un fichier *F.html* dans son SPIP_PATH et un fichier *F.php* dans un sous-répertoire balise de son SPIP_PATH. Le rôle de ce fichier PHP est de réceptionner les saisies demandées par ce formulaire, et éventuellement de ré-afficher celui-ci pour complément en cas de saisies invalides. C'est sans doute le type de contribution à l'espace public la plus difficile à réaliser, car la mécanique sous-jacente exige deux passes d'exécution de PHP dont il faut bien comprendre les rôles respectifs. Avant que ce mécanisme n'existe, la stratégie de développement de formulaire consistait à écrire des squelettes comportant des instructions PHP. Il est toujours possible de le faire, mais le résultat sera peu efficace car jamais mis en cache ; et il ne dispense pas de comprendre les deux passes de PHP intrinsèques au phénomène.

Le répertoire public contient le compilateur des squelettes. C'est une partie du code assez compliquée, mais depuis SPIP 1.8 elle bénéficie d'une interface de programmation qui rend ce compilateur totalement extensible sans exiger d'en comprendre tous les détails. La description la moins incomplète est [ici](#).

Le répertoire safehtml est en fait une bibliothèque indépendante de SPIP, fournissant des utilitaires de sécurisation des pages scriptables. Pour contribuer à cette partie, remonter directement à son [site](#).

Dernier point : la plupart des fichiers de SPIP sont utilisés via `charger_fonction` qui charge un fichier et appelle sa fonction homonyme censée y être définie. Il s'ensuit que le nom d'un fichier PHP doit être composé exclusivement de caractères acceptables pour un nom de fonction PHP : on évitera donc le signe moins, le point etc.

Règles de programmation

SPIP a démarré à l'époque où PHP transformait automatiquement en variables globales les paramètres HTTP. Ce style de programmation suicidaire a été abandonné par PHP4, SPIP n'a quant à lui pas tout à fait fini sa mutation sur ce point. Bien que le code actuel ne suive donc pas toujours les règles qui vont suivre, il est demandé aux contributions à venir de les respecter dès à présent, sans attendre que SPIP cesse d'y déroger ici ou là. On pourra lire attentivement le fichier `ecrire/articles.php`, le plus proche des spécifications qui suivent.

- Privilégier l'écriture par fonctions. La philosophie du logiciel libre est d'être utilisé dans un maximum de contextes différents, en conséquence le code doit être écrit dans une optique de réutilisation en dehors de son contexte initial de développement. L'écriture par fonctions ne référençant aucune variable globale et n'effectuant aucun appel à `echo` ou `print` garantit un chargement silencieux et un appel sans effets secondaires indésirables.
- Eviter au maximum l'utilisation de variables globales. Elles sont responsables de nombreuses failles de sécurité et d'impossibilités de réutilisation du code. Les alternatives à leur usage sont :
 - les constantes, qui ont l'avantage de signaler au lecteur que cette valeur ne changera pas pendant toute la durée du script ;
 - + les variables statiques, qui ont l'avantage de signaler au lecteur qu'il s'agit de valeur à longue durée de vie mais n'intéressant que la fonction qui la déclare.
- Ecrire du code ne produisant aucune erreur ni avertissement en mode

`error_reporting(E_ALL)`). Cela facilite la mise au point en cas de variable involontairement indéfinie. Si l'on a vraiment besoin d'exécuter du code en dehors de ce mode, utiliser l'artifice `@` en le limitant au maximum à la portion de code problématique, et prévoir un message d'erreur dans le journal, en utilisant la fonction `spip_log`.

- Commenter le *contexte*, pas le texte. Il ne sert à rien de paraphraser le nom de ses variables et fonctions, ni les fonctions PHP décrites dans son manuel : les commentaires comme *boucle sur le tableau de valeurs* devant un `foreach` ne font que grossir inutilement les fichiers. En revanche, il est souhaitable d'indiquer le type des arguments (PHP étant un langage dynamiquement typé, il se devine difficilement) et leur propriété supposée à l'entrée de la fonction (par exemple : non nul). Quand un bug difficile est corrigé ou anticipé, dire pourquoi le code initial était incorrect pour éviter qu'une réécriture ultérieure ne le réintroduise en croyant optimiser. Enfin, SPIP étant développé en français, éviter les termes absents des dictionnaires de cette langue afin de faciliter la compréhension à ceux dont ce n'est pas la langue maternelle.
- Nommer rationnellement les fonctions et variables. L'organisation du code de SPIP est plutôt fondée sur le découpage en répertoires dédiés que sur des règles de nommage strictes, mais on évitera néanmoins les incohérences comme le multi-linguisme à l'intérieur d'un nom. Les fonctions d'un même fichier tenderont à avoir un préfixe ou un suffixe commun, inspiré du nom du fichier.
- Tester sur un maximum de configurations. N'oubliez pas que SPIP doit fonctionner sur toute plate-forme, tant côté client que côté serveur. Il y a nécessairement plusieurs navigateurs sur votre machine, tester votre code sur au moins deux d'entre eux. Dans la mesure du possible, essayez également plusieurs hébergeurs. Lorsque une plate-forme force une écriture étrange, la mentionner explicitement, en précisant sa version et la date de l'essai.

Règles de présentation et d'écriture

SPIP est soucieux de la qualité typographique des pages qu'il envoie aux clients HTTP, il est donc lui-même soucieux de celle de ses propres fichiers. Il est recommandé de respecter les règles qui suivent, assez standard en programmation.

Présentation

- Le texte des fonctions doit être court, idéalement inférieur à la taille d'un écran standard, afin que l'intégralité de leur structure soit visible d'un seul coup d'œil.
- Le texte des fonctions doit être indenté, chaque bloc contenu à l'intérieur d'une paire d'accolades devant être renforcé d'une unité d'indentation supplémentaire, cela s'appliquant évidemment récursivement.
- Prendre le caractère de tabulation pour indenter, car il permet à chacune de choisir librement la profondeur d'indentation dans les options de son éditeur de textes.
- Lorsqu'une instruction PHP occupe plusieurs lignes, mettre une ligne blanche avant et après ;
- Les fragments HTML, qu'il s'agisse de transitions (`<?php` et `?>`) ou d'expressions PHP seront des blocs XHTML complets : éviter de mettre une balise ouvrante au début d'une fonction et sa fermante à la fin, cela ne permet pas de déplacer facilement cette paire en cas de besoin.

Typographie

- Lors de l'utilisation de parenthèses ou de crochets, éviter les espaces après l'ouvrant et avant le fermant.
- Dans les expressions PHP laisser un espace de part et d'autre des opérateurs binaires (`+`, `=`, `*`, `AND`, ...).
- Les opérateurs unaires (`!`, ...) doivent être collés au paramètre auquel ils s'appliquent.
- Par convention, lors d'un appel de fonction, il n'y a pas d'espace devant la parenthèse ouvrante :

« f(\$x) » et non « f (\$x) ». A contrario, et pour bien distinguer, on laisse un espace devant la parenthèse quand il s'agit d'une structure de contrôle intégrée au langage : « if (!\$x) » et non « if(!\$x) ».

- Les virgules et points-virgules sont suivis mais non précédés d'un espace.

L'espace des traducteurs

Voir chapitre suivant : Traduire SPIP

L'espace des traducteurs accueille tous ceux qui souhaitent aider la communauté des utilisateurs de SPIP en participant aux travaux de traduction.

L'espace des traducteurs accueille tous ceux qui souhaitent participer :

- à la traduction de l'interface d'utilisation de SPIP,
- à la traduction de l'aide en ligne,
- à la traduction du site de la documentation officielle (spip.net),
- et même à la traduction de l'interface de notre outil de traduction !

N.B. Cet espace n'est pas destiné à répondre aux questions d'utilisation des interfaces dans différentes langues ou d'installation des fonctionnalités multilingues de SPIP, mais à coordonner le travail de *celles & ceux qui veulent participer activement aux travaux de traduction*. Pour les questions d'utilisation de SPIP, merci de consulter la documentation.

Liens utiles

- **L'interface de traduction en ligne**

Nous mettons à la disposition des traducteurs & traductrices une interface leur permettant d'effectuer les traductions directement au travers de leur navigateur habituel. Cette interface facilite de plus le travail de plusieurs traducteurs & traductrices, en même temps, sur la même langue.

Pour accéder à cette interface, vous devez vous inscrire (ci-contre).

N.B. Si vous utilisez cette interface, merci de vous inscrire sur la liste [spip-trad](#) pour coordonner vos efforts avec ceux des autres participants.

- **La liste de discussion spip-trad**

La liste *spip-trad* permet aux traducteurs & traductrices de se concerter et d'échanger des informations liées aux difficultés des traductions.

Si vous souhaitez participer aux traductions, nous vous conseillons de vous inscrire à cette liste.

- **Vue d'ensemble des traductions de la documentation**

Les traducteurs & traductrices qui travaillent sur la documentation peuvent, à partir de cette page, visualiser la liste des articles traduits, en cours de traduction, ou encore modifiés depuis qu'ils ont été traduits.

Pour accéder à cette page, vous devez vous inscrire (ci-contre).

Contribuer au développement de SPIP

Quelques règles

Si vous voulez contribuer à la programmation de SPIP, l'idée la plus importante à retenir est la suivante : *vous arrivez sur un projet qui est déjà fonctionnel*. Ce projet est muni d'un ensemble de règles qui, toutes arbitraires qu'elles peuvent paraître, assurent sa cohérence. Ces règles n'ont pas besoin d'être énoncées explicitement pour exister : certaines sont clairement visibles après un examen plus ou moins détaillé du code, et les règles tacites doivent être respectées au même titre que les autres.

Il est formellement conseillé de bien suivre ces règles. Ce respect n'a pas à être lié ou non à vos goûts personnels : il permet de garder sa cohérence et son unité au projet, et de le conserver aussi lisible qu'il l'était auparavant. N'oubliez pas que d'autres personnes que vous sont amenées à lire, comprendre, voire modifier votre code.

Par exemple, il est évident que les fonctions SPIP sont écrites sous la forme `ma_fonction()`. Dans le cadre de ce projet, il serait donc parfaitement déplacé d'ajouter des fonctions en les écrivant `MaFonction()` - même si dans l'absolu cette forme n'est pas plus critiquable que l'autre.

Tout ceci n'empêche pas évidemment de critiquer une règle et d'en proposer une meilleure, le cas échéant. N'hésitez pas à le faire ; mais il doit y avoir de véritables raisons à cela.

Enfin, toute règle souffre des exceptions. Mais celles-ci doivent avoir une véritable justification, non uniquement la paresse du programmeur ; elles doivent être le plus rares possible. Notamment, garder à l'esprit que le « provisoire » a souvent tendance à devenir définitif quand personne n'a envie de le corriger ; or il est logique et juste que tout programmeur soit responsable de la finition de son propre code, mais non de celui des autres.

Règles de présentation et d'écriture

Les règles qui suivent sont communes à un nombre plus ou moins grand de langages de programmation : au minimum tous les langages présentant une syntaxe similaire à PHP (c'est-à-dire, outre PHP lui-même, C, C++, Java...).

Ces règles sont communément acceptées, de façon aussi naturelle que les règles de présentation et de typographie d'un texte en langage naturel ; d'ailleurs elles sont fréquemment similaires.

Présentation

- Le code doit être espacé et indenté de manière à mettre en valeur sa structure et la séparation entre les différents blocs logiques (fonctions notamment). L'espacement et l'indentation doivent être suffisants pour rendre la structure compréhensible dès le premier regard ; ils ne doivent pas être excessifs. On doit y apporter le même soin qu'à la division en paragraphes d'un texte en langage naturel.
- L'indentation sera faite de préférence avec le caractère de tabulation. Cela permet de choisir librement la profondeur d'indentation dans les options de son éditeur de texte, tout en n'imposant pas ce choix aux autres développeurs.
- Tout bloc contenu à l'intérieur d'une paire d'accollades sera indenté d'une et une seule tabulation. De même, récursivement, pour les sous-blocs : ajout d'une et une seule tabulation supplémentaire à chaque entrée dans un niveau de profondeur supplémentaire. Cette règle vaut bien aussi pour la déclaration de fonctions.

- Le code qui ne fait pas partie d'une fonction ne doit pas être indenté.
- L'utilisation des transitions PHP-HTML (<?php et ?>) doit être minimisée. L'éviter lors qu'il s'agit d'afficher uniquement de petits morceaux de HTML. Ne pas oublier qu'un petit morceau de code PHP inséré au milieu d'un océan de HTML est invisible sans un examen très attentionné.

Typographie

- Lors de l'utilisation de parenthèses ou de crochets, il ne faut pas laisser d'espace après la parenthèse ouvrante ni avant la parenthèse fermante.
- Lors de l'utilisation d'opérateurs binaires (+, =, *, AND, ...), il faut laisser un espace de part et d'autre de l'opérateur. Exception manifeste dans cette phrase, où les opérateurs sont mentionnés et non utilisés en tant que tels.
- Les opérateurs unaires (!, ...) doivent être collés au paramètre auquel ils s'appliquent.
- Par convention, lors d'un appel de fonction, il n'y a pas d'espace devant la parenthèse ouvrante : « f(\$x) » et non « f (\$x) ». A contrario, et pour bien distinguer, on laisse un espace devant la parenthèse quand il s'agit d'une structure de contrôle intégrée au langage : « if (!\$x) » et non « if(!\$x) ».
- Les virgules et points-virgules sont suivis mais non précédés d'un espace.

Règles de programmation

Réfléchir

Avant de programmer une nouvelle fonctionnalité, réfléchir...

- méthodes et algorithmes utilisés pour l'implémentation : légèreté, performance, robustesse (ne pas hésiter à faire quelques calculs grossiers pour valider les choix) ;
- adéquation au projet : portabilité, sécurité, souplesse ;
- implications sur les autres fonctionnalités : modifications et ajouts à faire sur les fonctionnalités existantes ;
- place « naturelle » pour cette fonctionnalité dans le projet : en matière d'interface, de fichiers...

Ne pas négliger la factorisation ou mise en commun du code (par des fonctions, notamment dans des fichiers à inclure). Eviter par contre le plus possible les fichiers inclus contenant du code hors fonctions (sauf lorsque c'est « naturel » et voulu).

Nommer

- *Variables et fonctions :*

Quel que soit le projet, le nommage doit rester homogène pour que le code soit facile à lire. Ainsi, sous SPIP, les noms de variables et de fonctions seront en minuscules ; les noms composés, de la forme `variable_composee`.

D'une manière générale, les noms seront ni trop brefs, ni trop longs ; ils seront suffisamment explicites. Cette règle est particulièrement importante pour les variables globales, qui peuvent être partagées entre plusieurs fichiers et de nombreuses fonctions. Pour les variables locales (i.e. à une fonction), la règle est plus souple. Notamment, on peut employer des variables d'une lettre, par exemple pour obtenir des expressions plus compactes. Remarquer que dans tous les langages de programmation, un certain nombre de lettres sont par tradition associées à certains usages (exemples : \$i, \$j pour des compteurs de boucles, \$n pour un dénombrement, \$t pour un instant ou une durée en secondes...). Ne pas détourner ces conventions permet à vos lecteurs d'être plus vite dans le bain.

- *Fichiers :*

Pour des raisons historiques, les fichiers à inclure dans l'espace public seront appelés *inc-fichier.php3*. Dans l'espace privé, ce sera *ecriture/inc_fichier.php3* (noter le tiret bas à la place du tiret normal). Les fichiers de l'espace public appelés par redirection HTTP depuis l'espace privé sont appelés

spip_fichier.php3. Tous les autres fichiers ont un nom qui ne commence ni par "inc", ni par "spip".

Tester

Une fois une modification importante apportée, il est bon de la tester soi-même, sans attendre que quelqu'un d'autre le fasse à sa place. Dans le cadre de SPIP, cela veut dire vérifier que le programme marche de manière correcte sur un certain nombre d'hébergeurs (par exemple : Altern, Free...) et de configurations (par exemple : différentes versions de PHP, de MySQL, restriction plus ou moins grande des droits d'accès aux répertoires...) ; mais aussi qu'un certain nombre de situations parmi les plus courantes (dans le cas d'une interface graphique notamment) sont traitées correctement.

Partager vos modifications

Une fois que vous êtes satisfait de votre modification du code, il est grand temps d'en parler avec les autres développeurs de SPIP, et de voir s'il mérite d'être intégré à la distribution officielle de SPIP... Rendez-vous sur la liste de diffusion spip-dev. A bientôt !

Traduire SPIP

Comment traduire SPIP ?

Si vous voulez traduire SPIP, il y a deux manières de procéder :

Avec l'interface de traduction en ligne

La méthode qui a notre préférence est de procéder en ligne sur le site de la documentation de SPIP, à **travers l'interface de traduction**. Pour vous servir de cette interface il est nécessaire que vous vous [inscriviez d'abord comme rédacteur](#) sur ce site. Ensuite, voyez si votre langue est déjà en cours de traduction, auquel cas il faudra vous joindre à l'équipe de traduction correspondante (dans l'espace privé, ou via la liste [spip-trad](#)) ; sinon, vous pouvez créer la langue en question et démarrer sans attendre.

Attention : lorsqu'une variable est présente dans une chaîne, elle est notée entre deux « @ ». Vous aurez, par exemple, à traduire la chaîne : 'Erreur dans la sauvegarde (@type@ @id_objet@) !'. Il ne faut pas modifier les caractères entre les @, car ils seront remplacés pour former la chaîne finale : « Erreur dans la sauvegarde (article 7) ».

C'est à peu près tout ce qu'il faut savoir pour traduire SPIP. Pour le reste, les qualités de la traduction sont celles qu'on attend de SPIP : lisibilité, accessibilité à un public non initié (éviter tant que possible le jargon), correction de la langue (éviter les fautes de frappe ou d'orthographe), etc. Et bien sûr, il faut comprendre et connaître SPIP un minimum, et réfléchir au sens de ce que l'on traduit dans son contexte d'utilisation, car un faux sens est vite arrivé...

Il y a au total environ un millier de lignes à traduire (ça va plus vite qu'on ne pense...).

Hors connexion, directement dans les fichiers de langue

Si ce fonctionnement en ligne vous pose problème, et que vous préférez effectuer toute la traduction chez vous, hors connexion, vous pouvez le faire en téléchargeant SPIP et en travaillant à partir des fichiers de langue que vous éditez dans un simple éditeur de texte, puis que vous nous enverrez par mail pour intégration au [CVS](#).

La distribution de SPIP contient un répertoire `ecrire/lang/` dans lequel se trouvent, pour chaque langue, deux fichiers `spip_xx.php3` et `public_xx.php3`, où « xx » est le code de la langue. `spip_xx.php3` est utilisé par le logiciel lui-même et `public_xx.php3` par les squelettes de la distribution.

Ces fichiers sont organisés d'une manière relativement simple. Chaque chaîne de caractères disponible à la traduction y est représentée par une ligne du type :

```
'avis_chemin_invalide_1' => 'Le chemin que vous avez choisi',  
qu'il faut traduire en :  
'avis_chemin_invalide_1' => 'Li camino chi you have chost',  
etc.
```

Quelques particularités qui méritent votre attention :

- comme pour la traduction en ligne, lorsqu'une variable est présente dans une chaîne, elle est notée entre deux « @ ». Par exemple : 'avis_erreur_sauvegarde' => 'Erreur dans la sauvegarde (@type@ @id_objet@) !'. Il ne faut pas modifier les caractères entre les @, car ils seront remplacés pour former la chaîne finale : « Erreur dans la sauvegarde (article 7) » ;
- Les caractères accentués sont notés indifféremment en entités HTML (´) ou entités unicode (é) ;
- Le caractère apostrophe (') doit être précédé d'un backslash (\).

Traduire l'aide en ligne

[L'aide en ligne](#) est constituée d'une collection d'articles définis dans l'espace privé de ce site, et qui suivent un format précis. Pour la traduire adressez-vous à la [liste des traducteurs](#), qui sauront vous guider sur le format.

Conseils généraux pour la traduction de ce site

Le site www.spip.net a changé de présentation en janvier 2007. Voici quelques recommandations relatives à la façon dont ces nouveaux squelettes organisent et présentent le contenu de ce site.

Le rubriquage de www.spip.net

Le rubriquage du secteur français « [Documentation en français](#) » sert de référence. Il est visible *in extenso* à la page « [plan de site](#) ».

Il convient donc de le reproduire dans chaque secteur de langue. Cependant les secteurs peu traduits contiennent peu d'articles : mieux vaut alors ranger ceux-ci à la racine du secteur, sans s'embarasser du rubriquage.

Seules les rubriques principales bénéficient de logos. Ils ne contiennent pas de texte, ce qui permet de les utiliser dans toutes les langues. Il est donc conseillé de récupérer les icones de la version française pour les installer dans les rubriques correspondantes des autres langues. Le site n'utilise pas les « logos pour survol ».

Sur le site public, les rubriques et sous-rubriques sont classées par leurs numéros de titre, à l'exception des sous-rubriques de « [L'espace des traducteurs](#) » qui sont classées par langue.

Classement des articles dans les rubriques

Il y a deux façons de « classer » les articles sur www.spip.net :

- (par num titre) dans certaines rubriques, il est nécessaire de « forcer » le classement des articles selon une certaine logique (par exemple dans les [tutoriels](#) et le [manuel de référence](#)), qui est celle d'une explication qui se déroule sur plusieurs articles successifs ; il suffit de numéroter les titres des articles, avec un numéro suivi d'un point et d'un espace (« 1. Premier article », « 2. Deuxième article »...);

Les traducteurs et traductrices de la version espagnole ont adopté une numérotation de 10 en 10 très pratique : « 10. Premier article », « 20. Deuxième article ». Il est conseillé de procéder ainsi, cela facilitant l'insertion d'articles à l'intérieur de ce classement par la suite.

- (par date inverse) dans les rubriques qui ne nécessitent pas une lecture « à la suite » des articles (FAQ, trucs et astuces... et « [Évolutions et mises à jour](#) »), on ne numérote pas les articles, ce qui permet de présenter en haut de liste les derniers articles publiés, les plus anciens en bas.

Dans les squelettes, la présentation s'adapte automatiquement selon la présence ou non de la numérotation.

Rythmer la présentation de la page d'accueil

Le site est doté d'une page d'accueil principale et d'une page d'accueil par langue. Celles-ci ne sont pas qu'une simple liste de rubriques et d'articles : elles contiennent des pavés qui rythment l'interface et mettent en valeur certains éléments, de façon à présenter SPIP, mais aussi les ressources et les nouveautés dans chaque langue.



Le premier texte affiché est celui de la rubrique-secteur de la langue. Il présente SPIP et l'objet de ce site. Pour le modifier : « modifier la rubrique », champ « texte ».

Référencement de ressources externes

Les principaux sites de référence sont présentés dans les rubriques du secteur français « [Documentation en français](#) », et sont visibles à cette page « [Liens utiles](#) ». Bien qu'ils soient principalement francophones, on peut (mais ce n'est pas obligatoire) référencer ces sites en traduisant leurs descriptifs dans les autres secteurs de langue. Il est bien de compléter en référençant aussi les autres ressources disponibles dans chaque langue.

Chaque langue dispose d'une liste de discussion (spip-#lang@rezo.net). Ces listes sont signalées au cas par cas dans la rubrique « [Questions et réponses](#) » de la langue concernée. Celles-ci doivent être dotées du mot-clé « [spip-#lang@rezo.net](#) » pour être présentées en bonne place en page d'accueil. Les forums disponibles dans chaque langue sont référencés dans la même rubrique, et dotés du mot-clé « [Forum](#) » pour apparaître en lien dans la barre de navigation.

Pour faire un lien dans le corps d'un texte, on préférera employer le raccourci SPIP correspondant au site référencé. Par exemple, plutôt que d'écrire [SPIP Contrib->http://www.spip-contrib.net], on écrira [->site57] (ce qui s'affichera ainsi : [SPIP - Contrib](#)).

Le glossaire

Les boucles, balises, filtres, etc. de SPIP sont listés sous forme de mots-clés, qui sont associés aux articles qui les documentent en français, afin de constituer un [glossaire](#) complet du langage SPIP. Dans les autres langues, ces mots-clés sont inutiles, le squelette du glossaire proposant automatiquement les traductions disponibles.

Traduction des images

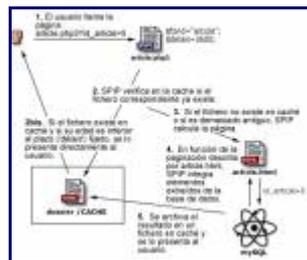
Fichiers image

Idéalement, on ré-utilisera le même fichier image que dans l'article d'origine. Par exemple, pour la traduction de l'article [3336](#), les images seront conservées telles qu'elles, c'est-à-dire appelées avec les mêmes raccourcis (<img2048|center>) que dans l'article d'origine. Pour qu'elles soient ainsi utilisables dans toutes les langues, on ne leur attribue ni titre, ni description.

Traduire le texte des images

Autant que possible, on évitera de créer des images contenant du texte. Si une explication textuelle est nécessaire, mieux vaut la placer dans les champs prévus (« titre » et « description ») de façon à faciliter les traductions. Ainsi un schéma nécessitant des explications sera plus aisé à traduire s'il n'affiche que des numéros, et que les explications correspondant à ces numéros sont reportées dans le champ de description : on fait alors une copie de l'image pour en traduire le titre et la description directement dans l'espace privé, sans qu'il soit nécessaire de faire appel à un logiciel de retouche d'images.

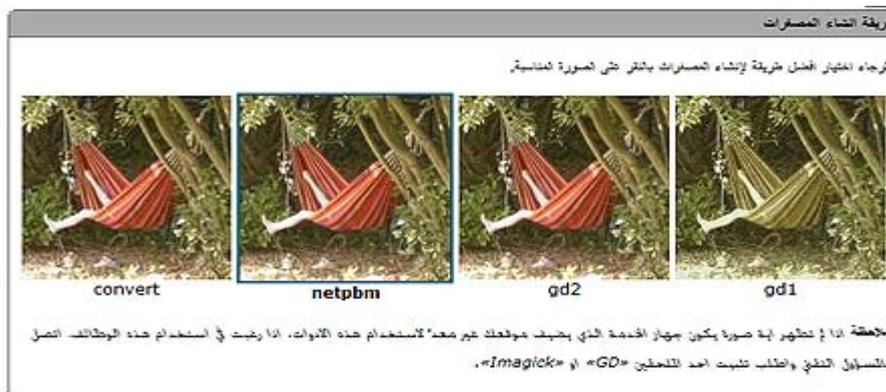
Cependant, il est parfois nécessaire de « traduire » certaines images, lorsqu'elles contiennent du texte. Pour traduire les textes sur des images de la documentation de SPIP, il faut utiliser un logiciel de retouche d'images.



Par exemple, le document ci-dessus a été réalisé avec [Fireworks](#) à partir de l'image en français (de l'article [877](#)), où les textes sont traduits en espagnol (pour l'article [93](#)). Vous pouvez reprendre ce document pour en faire plus rapidement une traduction vers une autre langue.

Des captures d'écran dans la bonne langue

De nombreuses illustrations de la documentation sont des captures d'écran de l'espace privé de SPIP. Lorsque le texte y est lisible, mieux vaut réaliser une nouvelle capture d'écran dans la même langue que celle de l'article qu'elle illustre, afin de faciliter la compréhension.



Ainsi la capture d'écran ci-dessus est utilisée dans l'article [3049](#) (en langue arabe), pour remplacer avantageusement celle de l'article [3024](#) (en langue française).

À propos des forums

Attention : les forums n'étant plus hébergés ce site, cet article n'est plus d'actualité. Il est cependant conservé pour archive, la partie concernant la modération étant toujours valable.

Les forums dans chaque langue

Les forums s'activent ici par l'intermédiaire d'une rubrique. En effet, dans tous les articles du site, on laisse la configuration « de base » des forums (c'est-à-dire activés et modérés à posteriori), simplement ils ne sont généralement pas affichés dans les squelettes ; mais cela laisse la possibilité, par la suite, de « jouer » avec les forums dans chaque page en fonction de futurs développements (par exemple : remettre le système de « notation » des FAQ).

Pour créer une rubrique de « forums », il suffit de lui attribuer, dans l'espace privé, le mot-clé « Forums ». Tous les articles publiés dans cette rubrique adopteront l'interface caractéristique des forums.

Sur spip.net, les forums sont thématiques. Le thème est le titre de l'article qui l'accueille :

- afin d'obtenir un classement « logique » des forums, numéroter les titres des articles (par exemple : « 1. Installation », « 2. Utilisation de l'espace privé »...)
- ne pas hésiter, dans le texte de l'article, à décrire en un ou deux paragraphes le sujet traité dans ce forum.

N.B. La rubrique des forums de la langue sont affichées dans la colonne de gauche de toutes les pages de rubriques.

Modération des forums

Lorsque vous créez l'article de forum, vous êtes automatiquement indiqué comme auteur de l'article (fonction SPIP de base). Si vous laissez ainsi, vous recevrez un mail à chaque message posté dans le forum de cet article. Cela peut s'avérer pratique pour suivre la vie du forum ; cela peut aussi devenir pénible si le forum est très actif (plusieurs messages par jour). À vous de déterminer si vous voulez recevoir ces mails de suivi des forums.

Pour ne plus recevoir ces mails, il vous suffit de vous retirer en tant qu'auteur de l'article. N.B. Selon le même principe, il suffit d'ajouter en tant qu'auteur de l'article toute personne qui souhaite recevoir ces mails.

D'expérience, nous savons que les forums de SPIP deviennent rapidement très très fournis. La nouvelle présentation est destinée à les rendre plus facilement consultable (notamment pour la recherche d'une question déjà posée). Pour faciliter la fonction d'archive de documentation de ces forums (j'ai un problème, je vais regarder si la question a déjà été posée et résolue), deux points de modération sont recommandés :

- les questions (les messages de premier niveau, qui lancent un fil de discussion) *doivent* avoir un titre explicite (de cette façon, le grand tableau qui récapitule toutes les questions est lisible, sinon il y a des threads dont on ne sait pas de quoi ils causent...) ; les questions qui n'ont pas de titre doivent être effacées (il y a un gros warning en gras en haut de page dans la page pour poster le message, alors, hein...) ; N.B. cela ne s'applique pas aux réponses aux questions, puisque ceux-ci adoptent automatiquement la mention « > La question », suffisante pour naviguer ;
- vérifier que les questions sont posées dans le bon forum (d'expérience, on sait que certains utilisateurs postent n'importe où) ; là encore, sucrer les messages qui ne sont pas au bon endroit.

Lorsqu'on supprime un message pour ces deux raisons, si la question est intéressante (formulation claire, sujet encore inédit dans le forum), on peut reposer un message avec le titre corrigé ou replacé dans le bon forum soit-même ; dans ce cas, il est conseillé d'indiquer qu'il s'agit d'un repostage, en citant le message d'origine en tant que citation (<html><quote>...</quote></html>).

Pour le reste, les forums consacrés à SPIP ne posent généralement pas de problèmes de modération, mais plus d'animation. En gros, il y a deux types de messages à surveiller :

- les messages qui débinent un hébergeur ou un produit commercial ; ce genre de messages, quasi systématiquement, nous attirent des menaces de poursuites ; si le message vous semble directement diffamatoire (genre : « l'hébergeur machin-online, c'est de la merde »), sucez-le... ne pas hésiter à demander à l'auteur du message de reformuler en une critique argumentée (« chez machin-online, l'utilisation de SPIP est difficile ou impossible pour telle et telle raisons... »), ce qui de toute façon sera plus utile aux autres participants ;
- les messages proposant une solution technique à base de PHP introduisant un trou de sécurité dans les sites (genre, attaquer la base de données avec des mots de passe directement dans le squelette) ; au minimum, poster immédiatement un message à la suite, avec des avertissements en gras, cligotants et en rouge vif pour expliquer que ce bout de code ne doit surtout pas être utilisé.