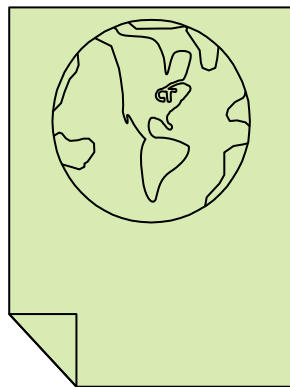


Création de Site Internet Technologies SPIP



Sommaire

| | |
|--|----|
| Sommaire | 2 |
| I. Les bases de la création de site | 4 |
| 1.1 Internet | 4 |
| 1.2 Fournisseur d'accès internet | 5 |
| 1.3 Navigateurs..... | 6 |
| 1.4 Les langages et format de donnée | 7 |
| 1.5 Serveur & Client..... | 8 |
| II. Les étapes de la conception d'un site internet | 9 |
| 2.1 Les étapes de la création..... | 10 |
| 2.2 Organisation et réflexion..... | 11 |
| 2.3 Documents du projet | 12 |
| III. Le HTML..... | 13 |
| 3.1 Définition | 13 |
| 3.2 HTML, un langage à balises..... | 13 |
| 3.3 Comment utiliser les balises HTML ?..... | 14 |
| 3.4 Imbrication des balises..... | 14 |
| 3.5 Notion d'attribut | 15 |
| 3.6 Espaces, saut de ligne et tabulations | 15 |
| 3.7 Commentaires..... | 15 |
| 3.7 Structure du document HTML | 15 |

| | | |
|------|---|----|
| 3.8 | Déclaration du type de document | 16 |
| 3.9 | Niveaux de titre | 16 |
| 3.10 | Utilisation de tableaux..... | 17 |
| 3.11 | Présentation des ancrages..... | 19 |
| 3.11 | Comment afficher des images sur une page web? | 20 |
| 3.12 | Editeur de document HTML..... | 23 |
| IV. | Solution LAMP (ou WAMP) | 24 |
| 4.1 | Serveur HTTP | 25 |
| 4.2 | Apache HTTP Server..... | 26 |
| 4.3 | MySQL..... | 27 |
| 4.4 | PHP: Hypertext Preprocessor | 29 |
| 4.5 | Introduction à EasyPHP | 32 |
| V. | Système de gestion de contenu | 36 |
| 5.1 | Fonctions communes | 36 |
| IV. | SPIP, Système de publication pour Internet | 41 |
| 6.1 | Présentation..... | 41 |
| 6.2 | Notions principales..... | 42 |
| 6.3 | Droits dans SPIP..... | 48 |
| 6.4 | Installer SPIP..... | 49 |
| 6.5 | Des boucles et des balises..... | 53 |
| | Sources | 64 |

I. Les bases de la création de site

1.1 Internet

Internet est le réseau informatique mondial qui rend accessibles au public des services comme le courrier électronique et le World Wide Web. Ses utilisateurs sont désignés par le néologisme « internaute ».

Techniquement, Internet se définit comme le **réseau public mondial utilisant le protocole de communication IP** (*Internet Protocol*). Internet ayant été popularisé par l'apparition du World Wide Web, les deux sont parfois confondus par le public non averti. En réalité, le web est une des applications d'Internet, comme le sont le courrier électronique, la messagerie instantanée et les systèmes de partage de fichiers poste à poste. Par ailleurs, du point de vue de la confidentialité des communications, il importe de distinguer Internet des **intranets**, les réseaux privés au sein des entreprises, administrations, etc., et des **extranets**, interconnexions d'intranets pouvant emprunter Internet. Même si historiquement la technologie utilisée pour internet émane de l'armée, l'idée altruiste d'internet est fortement liée à celles apparues plus tard que sont les logiciels libres.

Un **logiciel libre** est un logiciel dont la licence dite libre donne à chacun le droit d'utiliser, d'étudier, de modifier, de dupliquer, de donner et de vendre ledit logiciel. Richard Stallman a formalisé la notion de logiciel libre dans la première moitié des années 1980 puis l'a popularisée avec le projet GNU et la *Free Software Foundation* (FSF). Les logiciels libres constituent une alternative à ceux qui ne le sont pas, qualifiés de « propriétaires » ou de « privés ». Depuis la fin des années 1990, le succès des logiciels libres, notamment de **Linux**, suscite un vif intérêt dans l'industrie informatique et les médias. La notion de logiciel libre ne doit se confondre ni avec celle de logiciel gratuit (**freewares** ou *graticiels*), ni avec celle de **sharewares**, ni avec celle de domaine public.

De même, les libertés définies par un logiciel libre sont bien plus étendues que le simple accès au **code-source**, ce qu'on appelle souvent logiciel Open Source ou « à sources ouvertes ». Toutefois, la notion formelle de logiciel Open Source telle qu'elle est définie par **l'Open Source Initiative** est reconnue comme techniquement comparable au logiciel libre.

1.2 Fournisseur d'accès internet

Un **fournisseur d'accès à Internet (FAI)**, est un organisme (généralement une entreprise) offrant une connexion au réseau informatique Internet. Le terme anglais désignant un FAI est **Internet Service Provider (ISP)**. Beaucoup d'entreprises de télécommunications sont également des FAI. Avec l'explosion de la bulle Internet au début des années 2000, de nombreux FAI ont fait faillite, ou ont dû s'adapter pour survivre (restructuration, fusion, rachat).

Les différences entre les offres se font sur les tarifs, la bande passante (montante et descendante) offerte, et le support clientèle mais aussi sur les services ajoutés tel que l'hébergement de site web, l'accès à des boîtes de courrier électronique, l'accès à Usenet. Le particulier néophyte s'intéressera surtout à la facilité d'utilisation et à la documentation et se méfiera de la qualité souvent variable du support, les plus avancés lorgneront du côté de meilleur rapport prestations techniques/prix, le professionnel s'intéressera surtout à la tolérance aux pannes et demandera des garanties de service (les SLA : Service Level Agreement). Par ailleurs, d'autres différences techniques peuvent exister, notamment la disponibilité d'adresse IPv6 ou IPv4 fixes.

Les principaux FAI en France sont :

| | |
|---|---|
| <ul style="list-style-type: none"> ➤ Orange (France Telecom) | 49,3 % de parts de marché fin 2007. Et 6,4 millions d'abonnés dont 3,7 millions de clients haut-débit à une date antérieure non précisée. |
| <ul style="list-style-type: none"> ➤ Neuf Cegetel | 21,6% du marché fin 2007. Après le rachat de Club internet, Neuf Cegetel affichait plus de 3 000 000 d'abonnés ADSL. |

| | |
|----------|---|
| ➤ Free | 19,6 du marché fin 2007 et 2 900 000 de clients ADSL en mars 2008 dont 75% en dégroupage total. |
| ➤ Autres | 3,4% fin 2007. |

Un **nom de domaine** (ou **NDD** en notation abrégée) est un identifiant unique lié à une entité dont les ordinateurs sont reliés au réseau Internet. Le système est hiérarchique, permettant la définition de sous-domaine(s). Si le nom est libre, il peut être acheté auprès d'un registrar ou bien directement auprès du registre. Les labels ne doivent pas dépasser 63 caractères. Les noms de domaines peuvent donner lieu à des conflits juridiques, chaque partie revendiquant la propriété intellectuelle du nom litigieux. Depuis le 7 décembre 2005, il est possible de réserver un nouveau nom de domaine .eu auprès d'EURId. Les **registres** sont des bases de données hébergeant les noms de domaines sur Internet. Les **registres** (souvent appelés **NIC** pour *Network Information Center* même si leur rôle a beaucoup changé) sont des sociétés commerciales ou des associations qui ont pour vocation de maintenir une base de données unique par TLD (Top Level Domain, par exemple .com, .ch, .org, .info, .ci, .fr, etc.) où sont regroupées certaines informations sur les noms de domaine. Certains **registres** vendent directement aux utilisateurs, d'autres ne vendent pas les noms de domaines au public, ce sont alors les registrars qui le font.

Exemple d'extension

| | | |
|--------------------|--------------------|-------------------|
| .aero Aéronautique | .biz Business | .com Commercial |
| .coop Coopératives | .edu Education | .eu Europe |
| .info Information | .int International | .net Networks |
| .name Personal | .gov Government | .pro Professional |
| .org Organisations | .mil Militaire | .museum Museum |

1.3 Navigateurs

Un **navigateur web** est un logiciel conçu pour consulter le World Wide Web. Techniquement, c'est au minimum un client HTTP. Le terme **navigateur web** est inspiré de Netscape Navigator. D'autres métaphores sont ou ont été utilisées. Le premier terme utilisé était *browser*, comme en anglais. Par la suite, on a vu fureteur (surtout utilisé au Québec), butineur, brouteur, arpenteur, fouineur ou explorateur (inspiré de *Microsoft Internet Explorer*).

Pour la grande majorité des navigateurs sur le marché, l'interface utilisateur est composée d'une zone d'affichage éventuellement gérée sous forme d'onglets, d'une barre de menus déroulants, d'une barre d'outils et d'une barre d'état. La barre de menus abrite les favoris (ou marques pages), les commandes des fichiers (ouverture, fermeture), les options de configuration, etc.

Les navigateurs implémentant le HTML n'ont donc pas tous le même comportement, c'est pour cela qu'il est fréquent de voir des différences d'affichages entre différents navigateur, pour une même page.

Du fait de sa forte diffusion dans le monde le navigateur est une cible privilégiée des pirates désireux de nuire aux internautes.

L'internaute est menacé via son navigateur d'être victime des attaques suivantes :

- profilisation et analyse comportementale à des fins de revente marketing
- vol de données personnelles par accès à ses fichiers
- escroquerie par phishing
- crash de son ordinateur par bogue du navigateur

1.4 Les langages et format de donnée

Le **format des données** est la manière utilisée en informatique pour représenter des données sous forme de nombres binaires. C'est une convention (éventuellement normalisée) utilisée pour représenter des données, soit des informations représentant un texte, une page, une image, un son, un fichier exécutable, etc. Lorsque ces données sont stockées dans un fichier, on parle de **format de fichier**. Une telle convention permet d'échanger des données entre divers programmes informatiques ou logiciels, soit par une connexion directe soit par l'intermédiaire d'un fichier. On

appelle interopérabilité cette possibilité d'échanger des données entre différents logiciels.

Un **langage de programmation** est un code de communication, permettant à un être humain de dialoguer avec une machine en lui soumettant des instructions et en analysant les données matérielles fournies par le système, généralement un ordinateur. Le langage permet à la personne qui rédige un programme, de faire abstraction de certains mécanismes internes, généralement des activations et désactivations de commutateurs électroniques, qui aboutissent au résultat désiré.

L'activité de rédaction du code source d'un programme est nommée programmation. Elle consiste en la mise en œuvre de techniques d'écriture et de résolution d'algorithmes informatiques, lesquelles sont fondées sur les mathématiques. À ce titre, un langage de programmation se distingue du langage mathématique par sa visée opérationnelle (une fonction et par extension, un programme, doit *retourner* une valeur), de sorte qu'un « langage de programmation est toujours un compromis entre la puissance d'expression et la possibilité d'exécution. »

Exécution par le serveur HTTP (côté serveur) : PHP, ASP, JSP (issu de Java, basé sur des Servlets), Python, Perl, XSP (issu de XML, soutenu par Apache), CGI.

Exécution par le navigateur web (côté client) : JavaScript ou ECMAScript, VBScript, applets écrites en Java, ActionScript, XSLT.

Un **langage de requête** est un langage informatique utilisé pour effectuer des requêtes sur des bases de données ou autres systèmes d'information. La grammaire d'un langage de requête est donc adaptée à la structure des données interrogées.

1.5 Serveur & Client

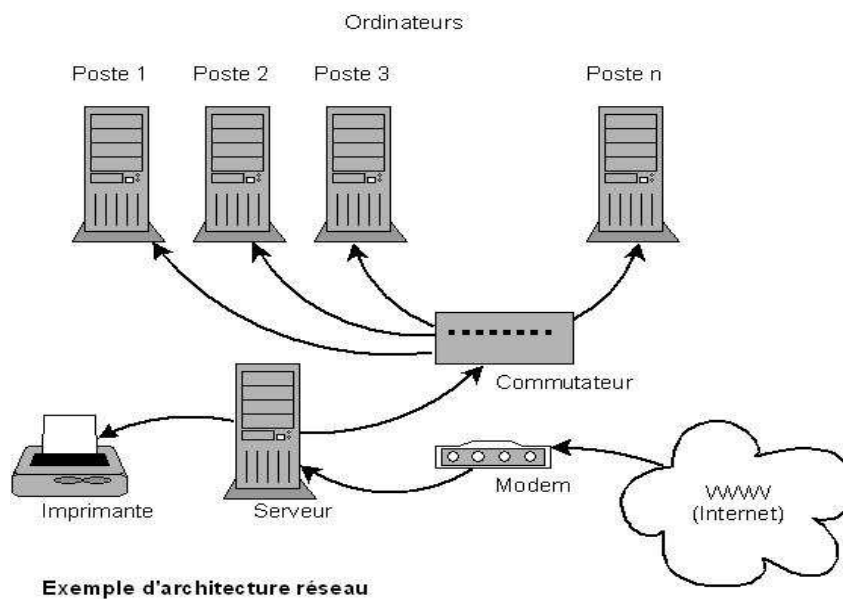
Au sens propre, un **serveur** est une entité qui effectue un service, ce terme est employé dans plusieurs domaines. Dans la restauration, un **serveur** est une personne travaillant dans un café ou un restaurant et qui sert les clients. En informatique, par analogie on appelle un **serveur** une machine ou un programme qui offre un service à un **client**.

Un **serveur informatique**, ou **serveur** lorsque le contexte s'y prête, est l'un des éléments participant au mode de communication client-serveur entre des logiciels : un logiciel dit « *client* » envoie une *requête* à un logiciel « **serveur** » qui lui répond, le tout suivant un protocole de communication.

Par extension, on désigne par serveur informatique l'ordinateur hébergeant de tels logiciels serveurs. Les logiciels clients s'y connectent à travers un réseau informatique. Les serveurs offrent des services qui permettent, par exemple, de stocker des fichiers, transférer le courrier électronique, héberger un site Web, etc. Il est possible pour un ordinateur ou un logiciel d'être client et serveur en même temps.

La connexion client-serveur utilise des protocoles de communication, comme par exemple TCP/IP, qui est le protocole le plus utilisé sur l'Internet.

Exemple de serveurs : Serveur central, Serveur d'application, Serveur d'impression, Serveur de base de données, Serveur de fichier, Serveur Web.



II. Les étapes de la conception d'un site internet

La création et la **conception d'un site web** est un processus qui implique quelques étapes simples au départ, mais pouvant sur un site abouti mettre en œuvre des

moyens et des compétences très importants. Un site web, cela peut être une simple page au format HTML, ou des milliers de pages proposant des animations, des services programmés en Java, PHP ou autre langage serveur, des formulaires supposant un traitement en JavaScript, ou AJAX. Il peut reposer sur des technologies de bases de données [p.ex. MySQL].

2.1 Les étapes de la création

En fonction du type de site, du contexte et des moyens disponibles pour le mettre en œuvre certaines de ces étapes sont optionnelles voire inutiles.

- Réflexion sur l'objectif du site, sa rentabilité, les moyens financiers à engager
- Réflexion sur le contenu : définition d'une arborescence, des pages, de la navigation, de la stratégie de référencement...

Une agence web peut-être mise à contribution dès cette étape pour apporter son expertise et se charger de plusieurs des tâches à venir.

- Réflexion sur l'autonomie souhaitée et le type de moyens humains pour faire la mise à jour.
- Dépôt d'un nom de domaine et choix d'un hébergeur.
- Établissement d'une structure de pages HTML ou XHTML.
- Mise au point d'une charte graphique.
- Choix et installation d'un Système de gestion de contenu (SPIP, wiki ...) ou choix d'un éditeur de site web de type WYSIWYG (exemples : NVU, Macromedia Dreamweaver, Microsoft FrontPage) ou texte (exemple : Bluefish, Emacs, ...).
- Création de pages via le Système de gestion de contenu ou l'éditeur. Les pages peuvent être créées individuellement ou reposer sur un système de gabarits.
- Intégration de la charte graphique via les feuilles de styles CSS.

2.2 Organisation et réflexion

- Quelles sont les personnes qui vont participer au projet ?

Il faudra veiller à avoir dans l'équipe des personnes dont le domaine est la technique (développeur, graphiste, ...) ainsi que des personnes qui connaissent suffisamment le domaine d'activité ainsi que l'activité de la structure : historique, projets, fonctionnement (président, chargé de projets,...).

- Quel est la cible du site internet et quel est sa particularité ?

La simple cible « tous les internautes » reste trop vague pour pouvoir créer un site internet adapté. Il est préférable de définir plusieurs cibles pour pouvoir dégager plusieurs particularités.

- Quels sont les informations essentielles à diffuser sur le site ?

Selon la cible, toutes les informations ne sont pas essentielles et au contraire certaines deviennent indispensables. Il existe par contre un assez grand nombre de classiques, souvent regroupé dans le menu. Présentation de la structure, Historique, Mission/Objectifs, actualités, contacts, liens utiles, ...

- Quelles sont les fonctionnalités principales souhaitées sur le site ?

Lors de cette partie les personnes à l'origine du projet vont s'entretenir avec les personnes qui maîtrise la technique pour savoir s'il est possible d'envisager tel ou tel fonctionnalité qui faciliterait tel ou tel activité. On peut évoquer par exemple : la newsletter, la saisie des information de mise à jour en ligne, des jeux intégrés, ... Tout interaction avec l'internaute ou contenu qui est adapté en fonction de certaines informations (dynamique).

- Qui s'occupera de la mise à jour du site et comment ?

Il est très important d'élucider cette question pour avoir une liberté sur le contenu et l'actualité de cet outil de communication.

- Quels sont les délais de réalisation ?

L'échange sur ce sujet devrait permettre de créer un planning prévisionnel qu'il faut s'efforcer de respecter. Des tâches doivent être attribuées aux personnes techniques (réalisation selon les choix fonctionnels effectués) ainsi qu'aux responsables qui devront fournir les informations sur le contenu selon les délais prévus.

- Comment va être diffusée l'adresse internet du site ?

Cette question peut être traitée en même temps que les questions de partenariats car un bon moyen de référencer un site internet sur les moteurs de recherches est d'avoir un grand nombre de sites qui pointent vers votre propre site. On peut également penser aux moyens classiques : radio, presse locale, tracts, magazines spécialisés...

2.3 Documents du projet

La réalisation technique passe par un cycle de tâches où certaines étapes sont indispensables pour assurer un maximum de réussite au projet. Cette organisation nécessite notamment la rédaction de plusieurs documents qui atteste de la bonne réalisation du projet et permet à tout le monde de suivre les étapes une par une.

Cahier des charges

Ce document est le plus difficile à réaliser et peut lors de projets importants demander plusieurs semaines de travail en groupe. Ce document commence par une introduction qui explique de façon générale le projet. Ensuite on y trouvera la liste des fonctionnalités demandées ainsi que le détail des phases de la réalisation de ces parties (analyse, développement, déploiement). On pourra également trouver une maquette visuelle. Grace à toutes ces informations, un planning détaillé sera inclus dans le document. Ce document est validé par les deux parties et il se peut que l'on réalise plusieurs versions avant de se mettre d'accord définitivement.

État d'avancement

Un document électronique devrait être fait pour suivre l'avancement de création des pages et des fonctionnalités à réaliser. Il permettra également de référencer les informations de test sur le site internet.

Documents de test

Une fois que le projet est entièrement réalisé et mis en ligne, les responsables doivent tester toutes les fonctionnalités pour s'assurer du bon fonctionnement. Un document inspiré du document d'état d'avancement devrait être rempli pour la validation finale.

Procédure de maintenance

Une liste de procédures de base doit être également fournie avec tous les fichiers sources du site. Ces procédures comportent par exemple : procédure d'archivage, procédure de sauvegarde, procédure spécifique, documents d'aide pour les fonctions avancées...

III. Le HTML

3.1 Définition

L'**Hypertext Markup Language**, généralement abrégé **HTML**, est le format de données conçu pour représenter les pages web. Il permet notamment d'implanter de l'hypertexte dans le contenu des pages et repose sur un langage de balisage, d'où son nom. HTML permet aussi de structurer sémantiquement et de mettre en forme le contenu des pages, d'inclure des ressources multimédias dont des images, des formulaires de saisie, et des éléments programmables tels que des applets. Il permet de créer des documents interopérables avec des équipements très variés de manière conforme aux exigences de l'accessibilité du web. Il est souvent utilisé conjointement avec des langages de programmation (JavaScript) et des formats de présentation (feuilles de style en cascade). HTML est initialement dérivé du Standard Generalized Markup Language (SGML).

3.2 HTML, un langage à balises

Le HTML n'est pas un langage de programmation. Il s'agit d'un langage permettant de décrire la mise en page et la forme d'un contenu rédigé en texte simple.

Une page HTML est ainsi un simple fichier texte contenant des **balises** (parfois appelées *marqueurs* ou *repères* ou *tags* en anglais) permettant de mettre en forme le texte, les images, etc.

3.3 Comment utiliser les balises HTML ?

Une balise est un élément de texte (un nom) encadrée par le caractère inférieur (<) et le caractère supérieur (>). par exemple « <H1> ». Les balises HTML ne sont pas sensibles à la casse, c'est-à-dire qu'elles peuvent être saisies indifféremment en minuscules ou en majuscules ! Les balises HTML fonctionnent par paire afin d'agir sur les éléments qu'elles encadrent. La première est appelée « *balise d'ouverture* » (parfois *balise ouvrante*) et la seconde « *balise de fermeture* » (ou *fermante*). La balise fermante est précédé du caractère « / » :

`<marqueur>` *Votre texte formaté* `</marqueur>`

A titre d'exemple, les balises `` et `` permettent de mettre en gras le texte qu'elles encadrent : `` **Ce texte est en gras** ``

Les balises HTML peuvent parfois être uniques : la balise `
` représente par exemple un retour à la ligne.

3.4 Imbrication des balises

Les balises HTML ont la particularité de pouvoir être imbriquées de manière hiérarchique afin de permettre le cumul de leur propriétés. En contrepartie le chevauchement de balises n'est pas toléré par le standard HTML. Voici un exemple de texte formaté avec des balises imbriquées :

`<i><u>`*Comment ça Marche*`</u>`, *encyclopédie informatique libre*`</i>`

L'exemple ci-dessus donne le résultat suivant :

Comment ça Marche, *encyclopédie informatique libre*

En contrepartie l'exemple ci-dessous n'est pas correct :

`<i>`*Comment ça Marche*`</i>`, **encyclopédie informatique libre**``

3.5 Notion d'attribut

Un attribut est un élément, présent au sein de la balise ouvrante, permettant de définir des propriétés supplémentaires. Les attributs se présentent la plupart du temps comme une paire « clé=valeur », mais certains attributs ne sont parfois définis que par la clé. Chaque balise peut comporter un ou plusieurs attributs, chacun pouvant avoir (aucune,) une ou plusieurs valeurs.

Voici un exemple d'attribut pour la balise <p> (balise définissant un paragraphe), permettant de spécifier que le texte doit être aligné sur la droite :

```
<p align="right">Exemple de paragraphe</p>
```

3.6 Espaces, saut de ligne et tabulations

Le langage HTML ne tient pas compte des espaces, des tabulations et des sauts de ligne (ci-après appelés ou plus exactement il considère une suite d'un ou plusieurs espaces/tabulations/saut de ligne comme un seul espace. Cela permet notamment d'indenter le code HTML pour plus de lisibilité, sans modifier l'apparence de la page HTML dans le navigateur.

3.7 Commentaires

Il est possible d'ajouter des éléments d'information dans une page web sans que ceux-ci soient affichés à l'écran grâce à un jeu de balises spécifique, appelé *balises de commentaires*.

```
<!-- Voici un commentaire -->
```

Les balises de commentaires permettent de mettre en commentaire du texte mais peuvent également servir à commenter du code HTML.

3.7 Structure du document HTML

Un document HTML commence par la balise <HTML> et finit par la balise </HTML>. Il contient également un *en-tête* décrivant le titre de la page, puis un *corps* dans lequel se trouve le contenu de la page. L'en-tête est délimité par les balises <HEAD> et </HEAD>. Le corps est délimité par les balises <BODY> et </BODY>.

Voici par exemple une page HTML minimaliste :

```
<HTML>
  <HEAD>
    <TITLE>Titre de la page</TITLE>
  </HEAD>

  <BODY>
    Contenu de la page
  </BODY>
</HTML>
```

3.8 Déclaration du type de document

Il est conseillé d'indiquer dans la page HTML le **prologue du type de document**, c'est-à-dire une référence à la norme HTML utilisée, afin de spécifier le standard utilisé pour le codage de la page. Cette déclaration se fait par une ligne du type :

```
<!DOCTYPE HTML PUBLIC "-//W3C//DTD HTML 4.0//EN">
<HTML>
  <HEAD>...</HEAD>
  <BODY>Contenu de la page</BODY>
</HTML>
```

3.9 Niveaux de titre

Le langage HTML définit 6 niveaux de titre (en anglais *heading*), afin de définir une structuration hiérarchique des paragraphes dans un texte :

| Balise | Effet Visuel |
|--------|--------------|
| H1 | Test |
| H2 | Test |
| H3 | Test |
| H4 | Test |

| | |
|----|------|
| H5 | Test |
| H6 | Test |

| Attribut | Valeur | Effet Visuel |
|----------|---------|---|
| ALIGN | LEFT | Texte aligné à gauche |
| | RIGHT | Texte aligné à droite |
| | CENTER | Texte centré |
| | JUSTIFY | Texte justifié |
| NOWRAP | | Interdit au navigateur d'effectuer un saut de ligne. |
| ID | | Attribue un identifiant à la balise. Cet attribut est notamment utile pour la programmation en JavaScript |
| LANG | | Spécifie un langage différent |
| CLASS | | Assigne une classe au contenu (pour les feuilles de style). |

3.10 Utilisation de tableaux

Il est souvent utile de présenter des informations mieux structurées qu'avec des listes. Les tableaux permettent de les afficher en lignes et en colonnes. Les tableaux sont définis comme étant des suites de lignes. Un tableau doit respecter les quelques règles suivantes :

- Le tableau est encadré par les balises `<TABLE>` et `</TABLE>`.
- Le titre du tableau est encadré par `<CAPTION>` `</CAPTION>`
- Chaque ligne est encadrée par `<TR>` `</TR>` (*Table Row*, traduisez par *ligne du tableau*).
- Les cellules d'en-tête sont encadrées par `<TH>` `</TH>` (pour *Table Header* : *En-tête de tableau*)

- Les cellules de valeur sont encadrées par <TD> </TD> (Table Data: Donnée de tableau)

Voici un exemple de tableau HTML :

```
<TABLE BORDER="1">
  <CAPTION> Voici le titre du tableau </CAPTION>
  <TR>
    <TH> Titre A1 </TH>
    <TH> Titre A2 </TH>
    <TH> Titre A3 </TH>
    <TH> Titre A4 </TH>
  </TR>
  <TR>
    <TH> Titre B1 </TH>
    <TD> Valeur B2 </TD>
    <TD> Valeur B3 </TD>
    <TD> Valeur B4 </TD>
  </TR>
</TABLE>
```

Ce code donne le résultat suivant :

| Voici le titre du tableau | | | |
|---------------------------|-----------|-----------|-----------|
| Titre A1 | Titre A2 | Titre A3 | Titre A4 |
| Titre B1 | Valeur B2 | Valeur B3 | Valeur B4 |

| Attribut | Balises auxquelles il s'applique | Valeur | Effet Visuel |
|----------|----------------------------------|---------|--------------|
| ALIGN | THEAD | CENTER | centré |
| | TBODY | LEFT | Gauche |
| | TH | RIGHT | Droite |
| | TR | JUSTIFY | Justifié |
| | TD | TOP | Au-dessus |
| VALIGN | CAPTION | BOTTOM | En-dessous |
| | THEAD | TOP | En haut |

| | | |
|-----------------------|----------------------------------|--|
| (alignement vertical) | TBODY TH TR TD | MIDDLE Au milieu BOTTOM En bas |
| BORDER=n | TABLE | Taille de la bordure |
| CELLPADDING=n | TABLE | Espacement de n pixels entre le contenu des cellules et la bordure |
| CELLSPACING=n | TABLE | Epaisseur de la grille intérieure |
| COLS=n | TABLE | Nombre de colonnes |
| COLSPAN | THEAD TBODY TH TR TD | Débordement des cellules sur les colonnes adjacentes |
| ROWSPAN | THEAD TBODY TH TR TD | Débordement des cellules sur les lignes adjacentes |

3.11 Présentation des ancrages

Les **liens hypertextes** (*ancrages*) sont des éléments d'une page HTML (soulignés lorsqu'il s'agit de texte) permettant aux internautes de naviguer vers une nouvelle adresse lorsque l'on clique dessus. Ce sont les liens hypertextes qui permettent de lier des pages Web entre elles. Ils permettent notamment de naviguer :

- vers un autre endroit du document
- vers un fichier HTML situé à un emplacement différent sur la machine qui héberge la page
- vers une autre machine

L'attribut principal des ancrages est *href*. Il s'écrit sous la forme suivante :

```
<a href="Adresse ou URL" > .. </a>
```

Lien externe

Un lien externe est un lien vers une page pointée par son URL

Par exemple : ` Comment ça marche? `

Lien local

On peut créer un lien vers une page située sur le même ordinateur en remplaçant l'URL par le fichier cible. Ce lien peut être fait de façon relative, en repérant le fichier cible par rapport au fichier source. Si le fichier cible est "index.html" situé dans le répertoire parent, son lien s'écrira : ` ... `

Ce lien peut aussi être défini de façon absolue, en écrivant l'adresse du fichier cible de façon locale: ` ... `

Les signets

Il est également possible de créer un signet dans une page, c'est-à-dire marquer un endroit précis d'une page pour s'y rendre par un lien hypertexte. Les signets se définissent grâce à l'attribut NAME ou ID. La syntaxe est la suivante :

```
<balise id="signet"> ... </balise>
```

L'appel d'un signet se fait de la manière suivante :

```
<a href="#signet"> ... </a>
```

Il est ainsi possible de permettre à l'utilisateur de se déplacer au sein d'une même page ou bien d'accéder à une section particulière d'une autre page :

```
<a href="url/nom_du_fichier.html#signet"> ... </a>
```

3.11 Comment afficher des images sur une page web?






La balise **IMG** du langage HTML permet d'insérer des images dans une page HTML. L'image peut être située sur le même serveur que la page dans laquelle elle est insérée mais également sur un autre serveur en spécifiant son URL complète. Seuls les formats d'images suivants sont acceptés en standard dans les spécifications du W3C :

| Type | |
|------------------------|---|
| Les images JPEG (.JPG) | Les images ayant un grand nombre de couleurs seront bien compressées, c'est-à-dire qu'elles prendront moins de place, donc nécessiteront un temps de chargement moindre) |
| Les images PNG | Leur taille est faible dans le cas d'images avec peu de couleurs avec des tons uniformes, ce format permet en outre d'avoir des images entrelacées (qui s'affichent progressivement) avec une profondeur de couleurs de 24 bits et des images dont on définit une couleur comme transparente. |
| Les images GIF | Elles possèdent les mêmes atouts que les images PNG, si ce n'est que le format GIF est limité à 256 couleurs maximum et que ce format n'est pas totalement libre. |

Exemple de balise d'une image :

```
<IMG SRC="url_de_l_image"
    ALT="Texte remplaçant l'image"
    TITLE="Texte à afficher">
```

| Attribut | Valeur | Résultat | Effet Visuel |
|----------|--|--|--------------|
| ALIGN | bottom center left middle top right | Alignement de l'image. | |
| ALT | | Texte alternatif au cas où l'image ne s'affiche pas. | |

| | | | |
|--------|---------------|--|---|
| BORDER | Nombre entier | Nombre de pixels de la bordure. La couleur de la bordure peut être définie par l'attribut LINK ou TEXT de la balise <BODY>. Par défaut l'attribut BORDER vaut 1 ce qui crée un petit cadre autour de l'image. Pour ne pas avoir ce désagrément pensez à le définir comme étant égal à 0. |  |
| HEIGHT | Nombre entier | Hauteur de l'image (en pixels ou en %). Lorsque cet attribut précise une dimension différente de la dimension originale du graphique, le navigateur la redimensionne dynamiquement, ce qui peut provoquer un affichage de moins bonne qualité. |  |
| HSPACE | Nombre entier | Nombre de pixels d'ajustement entre l'image et le texte adjacent (horizontalement). |  Texte |
| NAME | | Permet de définir un nom pour l'image. Cet attribut est notamment utile pour la gestion des images en JavaScript . | |
| SRC | URL | URL de l'image | |
| TITLE | | Texte alternatif au cas où l'image ne s'affiche pas. | |
| VSPACE | | Nombre de pixels d'ajustement entre l'image et le texte (verticalement). | Essai de texte  |
| WIDTH | Nombre entier | Largeur de l'image (en pixels ou en %). Lorsque cet attribut précise une dimension différente de la dimension originale du graphique, le navigateur la redimensionne dynamiquement, ce qui peut provoquer un affichage de moins bonne qualité. |  |

3.12 Éditeur de document HTML

Une page HTML est un simple fichier contenant du texte formaté avec des balises HTML. Par convention l'extension donnée au fichier est .htm ou .html, mais une page web peut potentiellement porter n'importe quelle extension. Une page web peut être construite à partir du plus basique des éditeurs de texte (une application *bloc-notes* par exemple), mais il existe des éditeurs beaucoup plus évolués. Les éditeurs WYSIWYG («*What You See Is What You Get*»), littéralement «ce que vous voyez est ce que vous obtenez») sont des éditeurs graphiques permettant de travailler sur une page web telle qu'elle sera affichée sur un navigateur à quelques détails près. Grâce à ce genre d'éditeurs il est possible d'ajouter des balises par simple clic et d'en modifier les attributs en éditant leur propriétés dans un formulaire. Pour autant, afin d'utiliser au mieux ce genre d'éditeur, une connaissance préalable du HTML est tout de même très utile.

Éditeurs WYSIWYG

Ses deux principaux avantages sont la facilité d'utilisation et l'observation immédiate du rendu graphique. Ses deux principaux désavantages sont le manque de maîtrise sur la qualité du document HTML produit et les risques d'incompatibilité avec des navigateurs non prévus par l'éditeur (plus récents, plus anciens, ou moins courants).

Liste d'éditeurs : OPEN BEXI HTML Builder, Adobe GoLive, Adobe Dreamweaver (anciennement Macromedia Dreamweaver), KompoZer, Microsoft Expression (anciennement Microsoft FrontPage), Mozilla Composer, Nvu.

Éditeurs en mode texte

Ces éditeurs éditent directement en langage HTML : le code y apparaît et y est traité comme du texte. Ils requierent donc une compétence dans ce langage. En contrepartie, ils permettent à un utilisateur compétent de s'assurer de la qualité du document produit.

Liste d'éditeurs : HTML Kit, Notepad++, Quanta+, Bluefish, Isoffing eWriter Wisto, Htmledit.

IV. Solution LAMP (ou WAMP)

LAMP est un acronyme désignant un ensemble de logiciels libres permettant de construire des serveurs de sites Web. L'acronyme original se réfère aux logiciels suivants :

- « **L**inux », le système d'exploitation ;
- « **A**pache », le serveur Web ;
- « **M**ySQL », le serveur de base de données ;
- « **P**HP » à l'origine, puis « **P**erl » ou « **P**ython », les langages de script.

Même si les auteurs de chacun de ces programmes ne se sont pas coordonnés pour construire des plateformes LAMP, cette combinaison de logiciels s'est popularisée du fait du faible coût de l'ensemble et de la présence de tous ces composants dans la plupart des distributions Linux.

Cet acronyme a été inventé par Michael Kunze qui l'a utilisé pour la première fois en 1998 dans le magazine allemand c't. L'article en question voulait démontrer qu'un ensemble de logiciels libres pouvait concurrencer les offres commerciales disponibles.

Architecture

Les rôles de ces quatre composants sont les suivants :

- Linux assure l'attribution des ressources aux autres composants ;
- Apache est le serveur web « frontal » : il est « devant » tous les autres et répond directement aux requêtes du client web (navigateur) ;
- MySQL est un système de gestion de bases de données (SGBD). Il permet de stocker et d'organiser des données ;

- le langage de script PHP permet la génération de pages web dynamiques et la communication avec le serveur MySQL.

Tous les composants peuvent être situés :

- sur une même machine ;
- sur deux machines, généralement Apache et le langage de script d'un côté et MySQL de l'autre ;
- sur de nombreuses machines pour assurer la haute disponibilité (répartition de charge et/ou *failover*).

Variantes

De même que Python ou Perl peuvent remplacer PHP puisqu'ils commencent par la même lettre, certaines combinaisons utilisent le *P* pour PostgreSQL (qui remplace MySQL) et font désigner les composants *mod_perl* ou *mod_python* par le *M*. D'autres acronymes sont apparus pour nommer des plateformes de serveurs internet :

- LAMA : *Linux Apache MySQL ASP*,
- WAMP : *Windows Apache MySQL PHP*,
- MAMP : *Macintosh Apache MySQL PHP*,
- SAMP : *Solaris Apache MySQL PHP*,
- LAPP : *Linux Apache PostgreSQL PHP*.

4.1 Serveur HTTP

Un **serveur** HTTP ou **démon** HTTP ou HTTPd (*HTTP daemon*) ou (moins précisément) serveur web, est un logiciel servant des requêtes respectant le protocole de communication client-serveur HyperText Transfer Protocol (HTTP), qui a été développé pour le World Wide Web.

Un ordinateur sur lequel fonctionne un serveur HTTP est appelé serveur web. Le terme « serveur web » peut aussi désigner le serveur HTTP (le logiciel) lui-même. Les deux termes sont utilisés pour le logiciel car le protocole HTTP a été développé pour le web et les pages web sont en pratique toujours servies avec ce protocole. D'autres

ressources du web comme les fichiers à télécharger ou les flux audio ou vidéo sont en revanche fréquemment servis avec d'autres protocoles.

EasyPHP est un package WAMP, une plateforme de développement Web, permettant de faire fonctionner localement (sans se connecter à un serveur externe) des scripts PHP. EasyPHP n'est pas en soi un logiciel, mais un environnement comprenant deux serveurs (un serveur web Apache et un serveur de bases de données MySQL), un interpréteur de script (PHP), ainsi qu'une administration SQL PhpMyAdmin. Il dispose d'une interface d'administration permettant de gérer les alias (dossiers virtuels disponibles sous Apache), et le démarrage/arrêt des serveurs. Il permet donc d'installer en une seule fois tout le nécessaire au développement local du PHP. Par défaut le serveur Apache crée un nom de domaine virtuel (car local) `http://127.0.0.1` ou `http://localhost`. Ainsi, quand on choisit "Web local" dans le menu d'EasyPHP, le navigateur s'ouvre sur cette URL et affiche la page `index.php` de ce site qui correspond en fait au contenu du dossier `www` d'EasyPHP.

4.2 Apache HTTP Server

Apache HTTP Server, souvent appelé **Apache**, est un logiciel de serveur HTTP produit par l'Apache Software Foundation. C'est le serveur HTTP le plus populaire du Web. C'est un logiciel libre avec un type spécifique de licence, nommée licence Apache. La fondation Apache (*Apache Software Foundation* ou ASF) a été créée en 1999 à partir du groupe Apache (*Apache Group*) à l'origine du serveur en 1995. Depuis, de nombreux autres logiciels utiles au World Wide Web sont développés à côté du serveur HTTP.

Apache fonctionne principalement sur les systèmes d'exploitation Unix (GNU/Linux, OSX, BSD et UNIX) et Windows. La version Windows n'est considérée comme stable que depuis la version 1.2 d'Apache. Apache est utilisé par de nombreux produits, dont Websphere d'IBM, ainsi que par Oracle Corporation. Il est également supporté d'une façon ou d'une autre par les outils de développement Borland Delphi et Kylix, ainsi que par des CMS comme Drupal.

Fonctionnalités

Apache est conçu pour prendre en charge de nombreux modules lui donnant des fonctionnalités supplémentaires : interprétation du langage Perl, PHP, Python et Ruby, serveur proxy, Common Gateway Interface, Server Side Includes, réécriture d'URL, négociation de contenu, protocoles de communication additionnels, etc.

Les possibilités de configuration d'Apache sont une fonctionnalité phare. Le principe repose sur une hiérarchie de fichiers de configuration, qui peuvent être gérés indépendamment. C'est notamment utile aux hébergeurs Web qui peuvent ainsi servir les sites de plusieurs clients à l'aide d'un seul serveur HTTP. Pour les clients, cette fonctionnalité est rendue visible par le fichier `.htaccess`.

Parmi les logiciels aidant la maintenance d'Apache, les fichiers de log peuvent s'analyser à l'aide de nombreux scripts et logiciels libres tels qu'AWStats, Webalizer ou W3Perl. Plusieurs interfaces graphiques facilitent la configuration du serveur.

4.3 MySQL

MySQL est un système de gestion de base de données (SGDB). Selon le type d'application, sa licence est libre ou propriétaire. Il fait partie des logiciels de gestion de base de données les plus utilisés au monde, autant par le grand public (applications web principalement) que par des professionnels, au même titre que Oracle ou Microsoft SQL Server.

MySQL AB a été acheté le 16 janvier 2008 par Sun Microsystems pour un milliard de dollars US.

Caractéristiques

MySQL est un serveur de bases de données relationnelles SQL développé dans un souci de performances élevées en lecture, ce qui signifie qu'il est davantage orienté vers le service de données déjà en place que vers celui de mises à jour fréquentes et fortement sécurisées. Il est multi-thread et multi-utilisateurs.

C'est un logiciel libre développé sous double licence en fonction de l'utilisation qui en est faite : dans un produit libre ou dans un produit propriétaire. Dans ce dernier cas, la licence est payante, sinon c'est LGPL qui s'applique. Ce type de licence double est utilisé par d'autres produits.

Systèmes d'exploitation supportés

MySQL fonctionne sur de nombreux systèmes d'exploitation différents, incluant AIX, BSDi, FreeBSD, HP-UX, Linux, Mac OS X, NetWare, NetBSD, OpenBSD, OS/2 Warp, SGI Irix, Solaris, SunOS, SCO OpenServer, SCO UnixWare, Tru64 Unix, Windows 95, 98, NT, 2000, XP et Vista.

Les bases de données sont accessibles en utilisant les langages de programmation C, C++, C#, Delphi / Kylix, Eiffel, Java, Perl, PHP, Python, Ruby et Tcl ; une API spécifique est disponible pour chacun d'entre eux. Une interface ODBC appelée MyODBC est aussi disponible. En Java, MySQL peut être utilisé de façon transparente avec le standard JDO.

Depuis le rachat de MySQL AB par Sun, MySQL est devenu officiellement la base de données à utiliser conjointement avec le langage de programmation Java, ce qui donne une notoriété supplémentaire au SGDB auprès des entreprises utilisant Java.

Utilisation

MySQL fait partie du quatuor LAMP : Linux, Apache, MySQL, PHP. Le couple PHP/MySQL est très utilisé par les sites web et proposé par la majorité des hébergeurs. Plus de la moitié des sites web fonctionnent sous Apache, qui est le plus souvent utilisé conjointement avec PHP et MySQL.

Wikipédia ainsi que de nombreuses autres entreprises et services utilisent MySQL, tels que Google, Yahoo!, YouTube, Adobe, Airbus, Alstom, Crédit agricole, Linden Lab (*Second Life*), RATP, URSSAF, AFP, Reuters, BBC News, Leader Price, Système U, Cap Gemini, Ernst & Young, Alcatel-Lucent et d'autres.

MySQL étant récent, il grossit au fil de ses versions. Depuis la version 5, il est possible d'utiliser le PL/SQL originellement développé par Oracle afin d'utiliser des procédures et fonctions stockées ainsi que des déclencheurs. Par ailleurs, MySQL supporte la

norme SQL2 (utilisation des JOIN), ce qui fait de lui un SGBD sûr puisque la conformité à cette norme garantira sa compatibilité avec les requêtes normalisées.

Son absence de support des transactions et d'une gestion de l'intégrité des tables automatique (sauf en utilisant certains moteurs comme InnoDB) ne lui permet pas d'être utilisée dans des applications à données particulièrement sensible comme, par exemple, dans les sociétés bancaires, cependant, ses performances étant généralement plus importantes que la majorité des autres systèmes concurrents, et son prix d'implantation nettement inférieur, lui permet d'obtenir un certain succès auprès des entreprises ayant besoin d'une base de données peu onéreuse et/ou performante.

4.4 PHP: Hypertext Preprocessor

PHP (acronyme récursif pour *PHP: Hypertext Preprocessor*), est un langage de scripts libre principalement utilisé pour produire des pages web dynamiques via un serveur HTTP, mais pouvant également fonctionner comme n'importe quel langage interprété de façon locale, en exécutant les programmes en ligne de commande. PHP est un langage impératif disposant depuis la version 5 de fonctionnalités de modèle objet complètes. En raison de la richesse de sa bibliothèque, on désigne parfois PHP comme une plate-forme plus qu'un simple langage.

Présentation

Le langage PHP est utilisé principalement en tant que langage de script côté serveur, ce qui veut dire que c'est le serveur (la machine qui héberge la page web en question) qui va interpréter le code PHP et générer du code (constitué généralement d'XHTML ou d'HTML, de CSS, et parfois de JavaScript) qui pourra être interprété par un navigateur. PHP peut également générer d'autres formats en rapport avec le Web, comme le WML, le SVG, le format PDF, ou encore des images bitmap telles que JPEG, GIF ou PNG.

Il a été conçu pour permettre la création d'applications dynamiques, le plus souvent dédiées au Web. PHP est très majoritairement installé sur un serveur Apache, mais peut être installé sur les autres principaux serveurs HTTP du marché, par exemple IIS. Ce couplage permet de récupérer des informations issues d'une base de données,

d'un système de fichiers (contenu de fichiers et de l'arborescence) ou plus simplement des données envoyées par le navigateur afin d'être interprétées ou stockées pour une utilisation ultérieure.

C'est un langage peu typé et souple et donc facile à apprendre par un débutant mais, de ce fait, des failles de sécurité peuvent rapidement apparaître dans les applications. Pragmatique, PHP ne s'encombre pas de théorie et a tendance à choisir le chemin le plus direct. Néanmoins, le nom des fonctions (ainsi que le passage des arguments) ne respecte pas toujours une logique uniforme, ce qui peut être préjudiciable à l'apprentissage.

Son utilisation commence avec le traitement des formulaires puis par l'accès aux bases de données. L'accès aux bases de données est aisé une fois l'installation des modules correspondant effectuée sur le serveur. La force la plus évidente de ce langage est qu'il est devenu au fil du temps un incontournable des offres d'hébergement.

Libre, gratuit, simple d'utilisation et d'installation, ce langage nécessite comme tout langage de réseau une bonne compréhension des mécanismes sous-jacents ainsi qu'une connaissance des problèmes de sécurité.

La version 5.3 permettra d'utiliser les espaces de noms, un élément fondamental de l'élaboration d'extensions, de bibliothèques et de frameworks structurés.

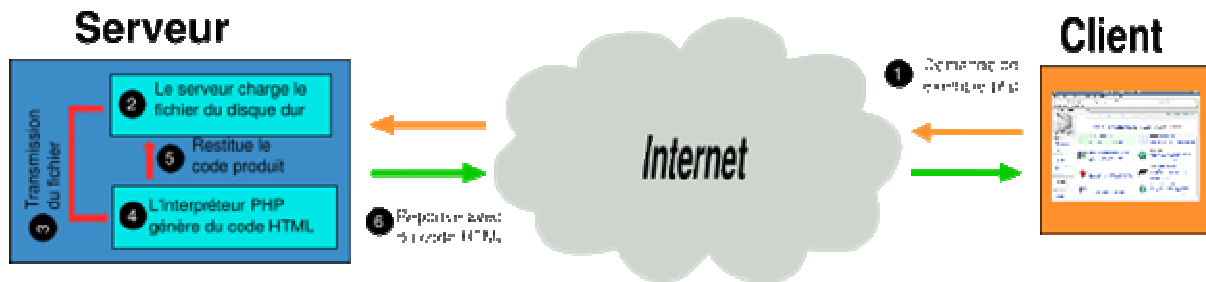
La version 6 introduira en interne la bibliothèque ICU donnant au langage la faculté de traiter Unicode de manière native.

Fonctionnement

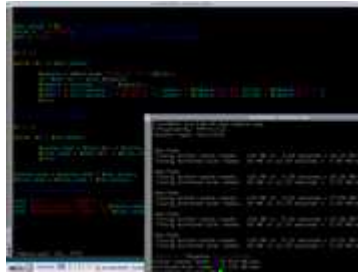
Sa syntaxe et sa construction ressemblent à celles des langages Java et Perl, à la différence que le PHP peut être intégré dans du code HTML. PHP appartient par ailleurs à la grande famille des descendants du C, dont la syntaxe est très proche.

Dans une utilisation Web, l'exécution du code PHP se déroule ainsi : lorsqu'un visiteur demande à consulter une page web, son navigateur envoie une requête au serveur HTTP correspondant. Si la page est identifiée comme un script PHP (généralement grâce à l'extension `.php`), le serveur appelle l'interprète PHP qui va traiter et générer le code final de la page (constitué généralement d'HTML ou de XHTML, mais aussi souvent de CSS et de JS). Ce contenu est renvoyé au serveur HTTP, qui l'envoie finalement au client.

Ce schéma explique ce fonctionnement :



Une étape supplémentaire est souvent ajoutée : celle du dialogue entre PHP et la base de données. Classiquement, PHP ouvre une connexion au serveur de SGBD voulu, lui transmet des requêtes et en récupère le résultat, avant de fermer la connexion.



Capture d'écran de vi en fond, affichant un script PHP, et à droite le résultat du script interprété par PHP

L'utilisation de PHP en tant que générateur de pages web dynamiques est la plus répandue, mais il peut aussi être utilisé comme langage de programmation en ligne de commande sans utiliser de serveur HTTP ni de navigateur.

Pour réaliser un script PHP exécutable en ligne de commande, il suffit comme en Perl ou en Bash d'insérer dans le code en première ligne le *shebang* : `#!/usr/bin/php` (`/usr/bin/` est le répertoire standard des fichiers binaires exécutables sur la plupart des distributions).

Il existe aussi une extension appelée **PHP-GTK** permettant de créer des applications clientes graphiques sur un ordinateur disposant de la bibliothèque graphique GTK. PHP possède un grand nombre de fonctions permettant des opérations sur le système de fichiers, la gestion des bases de données, des fonctions de tri et hachage, le traitement de chaînes de caractères, la génération et la modification d'images, des algorithmes de compression...

Le moteur de Wikipédia est écrit en PHP avec une base MySQL. Il s'intègre donc dans une plateforme LAMP.

4.5 Introduction à EasyPHP

Afin de faire fonctionner PHP, il est nécessaire à la base d'en télécharger les sources depuis un site spécialisé (par exemple PHP.net), puis de compiler celui-ci (ainsi que d'éditer les liens) afin de créer un fichier exécutable.

Ce processus demande des notions avancées en informatique, c'est pourquoi trois adeptes de PHP (Emmanuel Faivre, Laurent Abbal et Thierry Murail) ont mis au point un package (appelé *EasyPHP*) contenant 3 produits incontournables de la scène PHP :

- Le serveur Web Apache
- Le moteur de scripts PHP4
- La base de données MySQL
- Un outil de gestion de base de donnée graphique, Phpmyadmin

EasyPHP est ainsi un pack fonctionnant sous Windows permettant d'installer en un clin d'œil les éléments nécessaires au fonctionnement d'un site web dynamique développé en PHP.

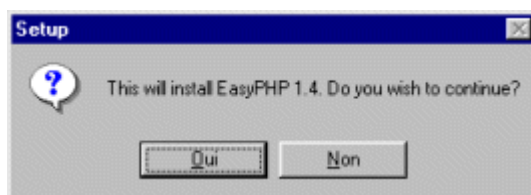
Récupérer EasyPHP

Le pack EasyPHP est disponible sur les sites suivants : www.easyphp.org

Il vous suffit dans un premier temps de télécharger la version la plus récente de EasyPHP.

Installer EasyPHP

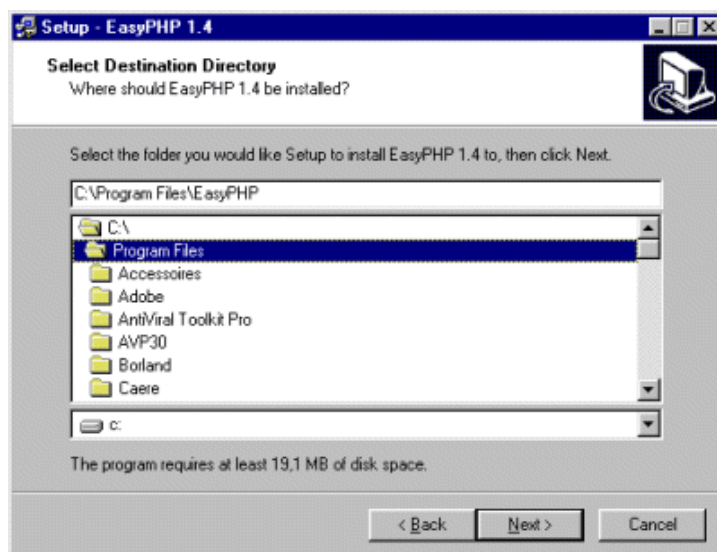
L'installation de EasyPHP est très simple, notamment avec l'apparition de la version 1.4 comportant un installeur automatique. Pour installer EasyPHP, il vous suffit dans un premier temps de double-cliquer sur le fichier téléchargé précédemment :



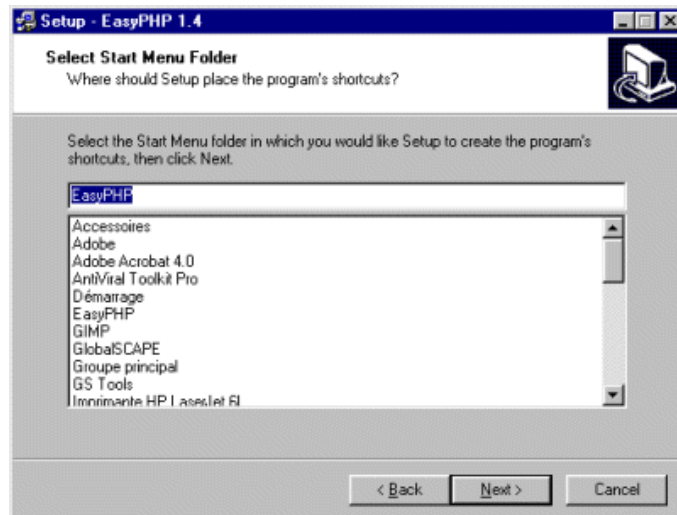
L'écran d'installation de EasyPHP suivant devrait apparaître, cliquez sur **Next** (Suivant) :



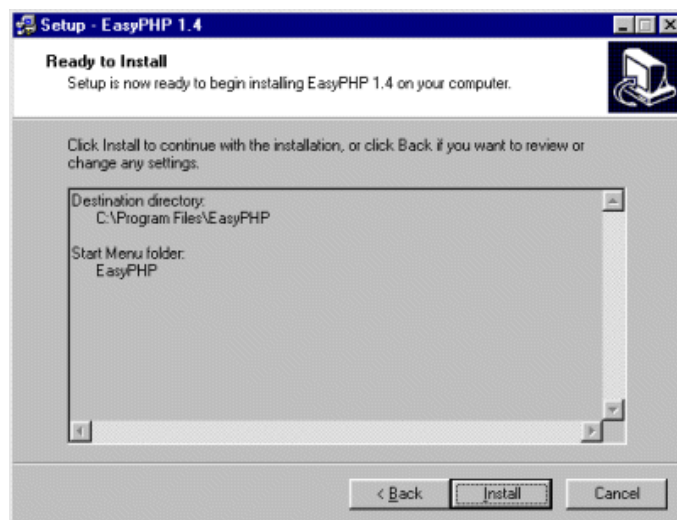
L'installeur va ensuite vous demander de préciser le répertoire d'installation :



Puis il va demander la création d'un groupe dans le menu démarrer.



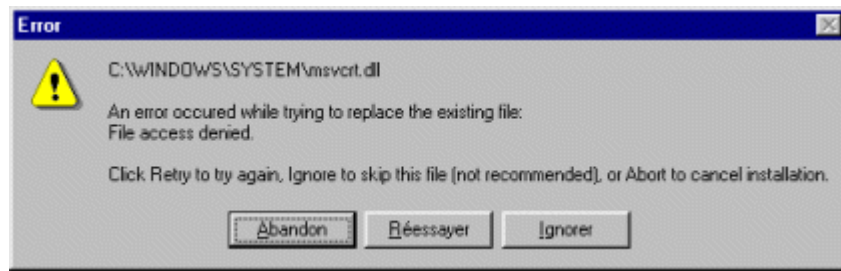
Et enfin il va vous récapituler les éléments de l'installation avant de procéder à la copie des fichiers.



Après la copie des fichiers, EasyPHP vous présente l'écran suivant indiquant que l'installation s'est déroulée correctement.



Il se peut lors de l'installation que l'installateur vous indique l'erreur suivante :



Cette erreur indique que la librairie msvcrt.dll n'a pu être copiée. La raison de cette erreur provient du fait que votre système Windows est actuellement en train d'utiliser cette librairie et ne peut donc l'écraser.

Pour y remédier, copiez cette librairie (par exemple dans c:\) sur votre disque dur, puis redémarrez en mode MS-DOS, puis tapez `copy c:\msvcrt.dll c:\windows\system`. Le système va vous demander de confirmer cet écrasement répondez "Oui" (Y ou O), redémarrez Windows et EasyPHP devrait fonctionner !

Démarrage de EasyPHP

Pour démarrer Apache, MySQL et PHP, il vous suffit de lancer EasyPHP à partir du groupe créé dans le menu démarrer :



Pour vérifier si EasyPHP fonctionne, il vous suffit de taper dans votre navigateur préféré : `http://localhost` ou `http://127.0.0.1` (ces deux adresses représentent votre machine locale).

Éditer votre site

Pour créer votre site web dynamique avec EasyPHP, il vous suffit de déposer vos créations dans le sous-répertoire /www de EasyPHP. Par exemple créez un fichier texte contenant le texte suivant :

```
<? phpinfo(); ?>
```

Puis renommez ce fichier en `phpinfo.php3` et déposez-le dans le sous-répertoire /www.

Vous pouvez désormais visualiser le résultat à l'adresse suivante :
<http://localhost/phpinfo.php3>

V. Système de gestion de contenu

Un **système de gestion de contenu** ou **SGC** ((**en**) *Content Management Systems* ou **CMS**) est une famille de logiciels destiné à la conception et à la mise à jour dynamique de site web ou d'application multimédia. Ils partagent les fonctionnalités suivantes :

- Ils permettent à plusieurs individus de travailler sur un même document ;
- Ils fournissent une chaîne de publication (*workflow*) offrant par exemple la possibilité de mettre en ligne le contenu des documents ;
- Ils permettent de séparer les opérations de gestion de la forme et du contenu ;
- Ils permettent de structurer le contenu (utilisation de FAQ, de documents, de blogs, de forums de discussion, etc.) ;
- Certains SGC incluent la gestion de versions.

Les SGC permettent de réaliser la gestion de contenu dans l'entreprise, il s'agit du ECM.

5.1 Fonctions communes

Pour arriver à offrir les fonctionnalités citées plus haut la conception se fonde assez souvent sur les mécanismes suivants :

Utilisation d'interface web

Au travers d'interfaces web, les SGC sont accessibles quel que soit le type de système d'exploitation au moyen d'un navigateur web. Ainsi, les utilisateurs n'ont pas besoin d'installer de logiciels spécifiques supplémentaires. Grâce aux standards du web, les SGC offrent donc un format de données lisible (HTML et ses dérivés), imprimable et stockable par tous, ce qui facilite l'échange et l'accessibilité des documents.

Séparation entre contenu et présentation

C'est un principe fondateur de la gestion de contenu :

- Le contenu est stocké le plus souvent dans une base de données, structurée en tables et en champs. C'est le contenu des champs de la base qui est créé/modifié par le rédacteur, et non pas la page elle-même. On parle de site « dynamique » ;
- La présentation est définie dans un gabarit. Le gabarit définit deux choses : la mise en page proprement dite - via les feuilles de style (parmi lesquelles les CSS, ou les règles de transformation XSLT), et la structuration des données, au moyen de standards tels que XML, ainsi que les informations extraites de la base de données (de même que l'endroit où celles-ci doivent être affichées et sous quelles conditions).

Édition de page simplifiée

Du fait du principe de séparation contenu/présentation, les rédacteurs peuvent se concentrer sur le contenu. L'édition des pages est considérablement simplifiée, et deux mécanismes sont proposés :

- Une interface WYSIWYG qui propose une interface graphique de mise en forme similaire à celle proposée par un traitement de texte ;
- L'utilisation de balises simplifiées, visant à mettre le texte en forme. Il en existe plusieurs types, dont le plus répandu est BBCode. On peut aussi citer des alternatives plus modernes telles que Markdown ou Textile. Les wikis qui ne sont pas WYSIWYG utilisent le wikitexte.

De multiples méthodes de rangement de l'information

La théorie de l'information nous apprend que, plus il y a d'informations, plus le désordre augmente. Cela signifie que, si l'on recherche une information spécifique, toutes les autres informations créent du bruit. Un SGC possède donc de multiples mécanismes de tris plus ou moins complexes comme :

- Les hyperliens, qui permettent de référencer les articles entre eux ;
- Un moteur de recherche sur le texte ;
- Des tris spécifiques (par date de modification, auteurs, liens, ...).

La majorité des SGC offrent la possibilité de catégoriser l'information, de l'indexer et d'utiliser des taxonomies pour améliorer les méthodes de recherche. On peut donc créer des catégories de contenus, des sections (ou rubriques), voire des mots clés favorisant l'indexation.

La multiplication des vues, des mécanismes de choix, diminue la profondeur de l'information par rapport à la page d'entrée en multipliant les chemins, et la rend donc plus accessible.

Gestion des droits

Par le biais de l'interface d'administration, le système offre la possibilité de gérer les utilisateurs et leurs droits. C'est à dire de leur donner accès ou non aux fonctionnalités du système. Les fonctionnalités sont multiples et variables selon la plate-forme, très basiques comme un droit de lecture/écriture sur un article, ou plus élaborées comme un droit de modération d'autres utilisateurs.

Gestion de versions successives et/ou concurrentes

Lorsque plusieurs personnes ou entités travaillent sur un même document, elle fournit un outil qui trace les évolutions, et permet de visualiser/gérer les modifications et les éventuelles situations conflictuelles. Grâce à elle, on peut également savoir qui a apporté *telle* modification et quand.

Travail collaboratif (groupware)

Plusieurs parties prenantes peuvent trouver avantage à travailler sur un espace commun privé, lié à une communication précise (suivi d'une relation client pour une entreprise et son fournisseur), tout en gardant une trace historique matérialisable (en général par la couleur) permettant de savoir qui a modifié quoi. Un logiciel précurseur dans le domaine a été Lotus WordPro, dont *Microsoft Word* reprit plus tard cette fonction.

Multiplication des sources de contenu [

Il s'agit de mutualiser les contenus de plusieurs organisations tout en présentant le contenu d'informations issues de sources différentes avec leur mise en page, comme le font les journaux qui diffusent les dépêches d'agences de presse. Ce mécanisme s'appelle généralement la syndication de site.

Commentaires devenant eux-mêmes sources d'information

La plupart des projets CMS libres fonctionnant sur le Web proposent de créer des forums associés aux articles pour laisser les visiteurs réagir. Ils sont la base du succès des sites de nouvelles en fil. On a pu constater sur des sites comme Slashdot que les commentaires des lecteurs apportaient eux-mêmes une valeur ajoutée à l'information.

L'amélioration qualitative se poursuit

Ces logiciels ont énormément amélioré leurs qualités :

La sécurité informatique, en particulier pour empêcher la prise de contrôle par des tiers, est désormais correcte voire excellente (prise en charge native de SSL, voire Kerberos) ;

La qualité du code (PHP, Python, PERL...) : il est de mieux en mieux indenté, commenté (donc plus clair), ce qui favorise l'identification des bugs et l'amélioration par la communauté ;

La qualité des documents informatiques publiés, la qualité du code HTML, JavaScript etc., le respect des normes ainsi que l'ergonomie.

La mise en conformité au standard XHTML et l'actualisation des méthodes de programmation des concepteurs avec des méthodes telles que l'Extreme programming ont permis une nette amélioration de la qualité des systèmes de gestion de contenu. Les initiatives menées par des individus comme Tristan Nitot ont permis de favoriser cette évolution.

Rentabilité

La mise en place d'un système de gestion de contenu simple pour une petite entreprise représente un investissement certain. Dans cette optique, l'entreprise doit considérer les points suivants afin d'évaluer si l'implantation d'un tel système sera rentable :

- étendue des modifications du site ;
- fréquence des modifications du site ;
- urgence des modifications du site.

Comparatif CMS : quel outil de gestion de contenu PHP choisir ?

Il existe **plus de 700 outils de gestion de contenu** (CMS pour "Content Management System") dans le monde. Et il est encore assez difficile de s'y retrouver dans cette profusion de CMS, d'autant plus qu'il n'y en a pas un seul qui sort du lot.

Le comparateur CMS Matrix (en anglais) présente, quant-à-lui, un comparatif exhaustif et détaillé des caractéristiques techniques et fonctionnelles des 740 systèmes de gestion de contenu qu'il recense. Le site CMS-Québec.com reprend, de manière moins approfondie, mais en français, le système de comparaison dynamique des principaux outils de gestion de contenu. Wikipédia met également à disposition une liste d'outils de gestion de contenu.

IV. SPIP, Système de publication pour Internet

Il s'agit d'un ensemble de fichiers, installés sur votre compte Web, qui vous permettent de bénéficier d'un certain nombre d'automatismes : gérer un site à plusieurs, mettre en page vos articles sans avoir à taper de HTML, modifier très facilement la structure de votre site...

Avec le même logiciel qui sert à visiter un site (Netscape, Microsoft Explorer, Mozilla, Opera...), SPIP permet de fabriquer et de tenir un site à jour, grâce à une interface très simple d'utilisation.



6.1 Présentation

SPIP a les caractéristiques suivantes...

Pour le(s) rédacteur(s) et administrateur(s)

- Une interface Web intuitive rend extrêmement simples la proposition d'articles et de brèves ainsi que la gestion éditoriale du site. De plus, des raccourcis typographiques permettent de mettre en forme un texte sans avoir à utiliser le langage HTML, rendant ainsi la contribution rédactionnelle accessible à tous, et aussi simple que l'écriture d'un e-mail.

Pour le webmestre

- L'aspect graphique et la navigation sont définis par des squelettes HTML (ou « formats types ») définissant chacun une « vue » (par exemple : une vue pour la page d'index, une autre montrant une rubrique et un résumé de son contenu, une troisième pour le détail d'un article, une quatrième pour le détail d'une brève). La façon dont est inséré le contenu rédactionnel du site dans ces pages est défini par un certain nombre de pseudo-tags HTML relativement faciles à maîtriser.

- SPIP ne restreint pas les possibilités graphiques et navigationnelles du site. Les squelettes HTML étant entièrement définis par le webmestre du site, il est possible de gérer certains éléments du site avec SPIP et le reste à la main ou même avec d'autres systèmes de publication (à condition que ces derniers soient aussi tolérants que SPIP, bien sûr).

Pour les visiteurs

- Un système de cache sur la partie publique du site accélère le site en évitant un grand nombre de requêtes à la base de données, et joue en outre un rôle de garde-fou contre les plantages de la dite base (fréquents sur des serveurs « chargés ») : dans ce cas, le site reste disponible de façon transparente, même si toute modification des contenus est impossible (y compris la contribution aux forums).

- Un moteur de recherche et d'indexation intégré à SPIP, s'il est activé par le webmestre, permet d'effectuer des recherches sur l'ensemble du contenu public du site.

6.2 Notions principales

La structure d'un site sous SPIP est construite sur une hiérarchie de *rubriques*. Il n'y a virtuellement pas de limite au nombre de rubriques : une rubrique peut contenir

autant de sous-rubriques que nécessaires, qui elles-mêmes contiennent des sous-rubriques, etc. On construit ainsi la structure de son site en imbriquant des rubriques et des sous-rubriques.

Les articles

L'objet principal permettant de publier des informations sous SPIP est l'*article*. On peut placer autant d'articles que nécessaire, dans n'importe quelle rubrique.

- La rédaction des articles est très simple, elle se déroule via une interface graphique sur le Web.
- Un article est constitué de plusieurs éléments qui permettent de le structurer : titre, surtitre, soustitre, descriptif, chapeau, texte principal, post-scriptum.
- Les règles de base de la typographie française sont appliquées automatiquement (espaces insécables avant les points d'interrogation, d'exclamation, etc.).
- Des raccourcis mnémotechniques facilitent l'enrichissement typographique, la création de liens hypertexte, de notes de bas de page... mettre en page un article sous SPIP est aussi facile que d'écrire un email.
- L'interface graphique permet d'inclure simplement des images dans les articles, et chaque article peut être signalé par son propre logo.
- On peut indiquer, pour chaque article, un ou plusieurs mots-clés.
- La date de mise en ligne se gère automatiquement (on peut cependant la modifier si nécessaire) ; une seconde date peut être associée à un article, par exemple pour indiquer une date de publication originale (par exemple, un article publié antérieurement dans un magazine papier).
- Redirections (articles « fantômes ») : SPIP permet de fabriquer des articles-fantômes, intégrés dans la structure du site et affichés dans le contenu des rubriques, mais qui en réalité renvoient vers une page dont l'adresse est spécifiée par le rédacteur (sur le même site, ou même sur un autre site). Cette fonction facilite le passage d'un site déjà existant vers SPIP, par l'intégration de contenus statiques préexistants.

Les brèves

En complément des articles, SPIP intègre un système de brèves, qui facilite la publication de courtes notes d'information, telles des revues de presse (ou des revues de Web).

- Afin de faciliter la structuration et le positionnement des brèves, on ne peut installer des brèves que dans les principales rubriques du site (les rubriques placées à la racine du site).
- La structure des brèves est simplifiée : un titre et le texte de la brève ; chaque brève peut être complétée très simplement d'un lien hypertexte.
- La gestion de la publication d'une brève est simplifiée (deux boutons : publier ou refuser).
- Chaque brève peut être signalée par son propre logo.
- L'administrateur du site peut décider de désactiver l'usage des brèves sur l'ensemble du site.

Les forums

SPIP intègre un système de forums.

- Les forums peuvent être associés aux articles (un forum par article), aux rubriques ou aux brèves. Le webmestre pourra programmer son interface pour que chaque article dispose de son propre forum, ou pour que plusieurs articles d'une même rubrique partagent le même forum, etc.
- SPIP permet de choisir entre plusieurs types de forums : les forums « libres » (modérés à posteriori, les contributions apparaissent immédiatement, les administrateurs peuvent éventuellement supprimer ensuite un message indésirable) ; les forums modérés à priori (les contributions n'apparaissent qu'après avoir été validées par un administrateur du site) ; les forums sur abonnement (chaque intervenant doit, pour pouvoir poster, d'abord indiquer son adresse email pour recevoir un mot de passe lui permettant de poster ses contributions).

SPIP intègre également un système de forums privés, consacré à la discussion entre les différents rédacteurs du site, et cela dans l'espace privé.

Les pétitions

Un article peut être transformé en pétition en ligne en quelques clics.

- Les pétitions de SPIP sont validées par email automatiquement : un signataire reçoit un message de confirmation qui permet de vérifier la validité des signatures.
- On peut configurer très simplement le type de pétition : ainsi imposer une seule signature par adresse email, imposer qu'un site Web soit indiqué dans la pétition (dans ce cas, la validité de l'URL est vérifiée automatiquement), accepter ou non des messages accompagnant les signatures.

Les statistiques

SPIP intègre un système très simplifié de statistiques, permettant d'évaluer la popularité des articles et des rubriques.

Les rédacteurs/administrateurs

Un site sous SPIP peut être géré par une seule personne, ou être réalisé par un groupe de rédacteurs.

- SPIP propose deux niveaux d'accès : les administrateurs, qui gèrent notamment la structure du site et la validation des articles, et les rédacteurs, qui proposent des articles.
- Le nombre de rédacteurs et d'administrateurs est illimité.
- On peut décider d'offrir aux utilisateurs du site public de s'inscrire pour devenir rédacteur (la procédure d'inscription est alors gérée automatiquement par SPIP).
- Chaque auteur peut se voir associer un logo personnel téléchargeable depuis l'interface (par exemple une photo d'identité).

Syndication

Les sites réalisés sous SPIP, phpNuke, ou d'autres systèmes, fournissent un fichier dynamique indiquant leurs dernières publications. SPIP peut analyser de tels fichiers et ainsi indiquer les nouveautés d'autres sites :

- on peut ajouter autant de sites syndiqués que l'on veut ;
- les sites syndiqués sont associés aux rubriques de son propre site ; ainsi, on peut associer à une rubrique thématique les liens vers des sites traitant du thème précis de la rubrique.

Interface graphique du site public

L'interface graphique du site public est très souple. Grâce à un langage très simple (mais propre à SPIP), on peut réaliser à peu près n'importe quelle interface graphique.

Il n'est en particulier pas nécessaire de connaître PHP et MySQL pour réaliser une interface graphique originale sous SPIP.

Cependant, le système de cache est totalement compatible avec PHP : le webmestre peut, s'il le désire, intégrer des fonctions PHP dans ses formats-types (squelettes), ou des passerelles CGI. On peut donc enrichir SPIP avec des scripts spécialisés pour compléter ou remplacer des fonctions manquantes (par exemple : compteur, moteur de recherche plus puissant, etc.).

Interface du site privé

La partie privée qui permet de gérer le site dispose d'une interface graphique complète, très simple d'utilisation.

- Cette interface s'adapte en fonction des activités de chaque rédacteur ou administrateur, et en fonction de l'activité du site. Ainsi chaque auteur a-t-il accès rapidement à ses propres articles, et les articles proposés à la publication sont signalés à tous les utilisateurs. De même l'interface est différente selon que l'on est rédacteur ou administrateur.
- Chaque utilisateur peut personnaliser son interface. Il peut choisir entre une interface simplifiée, qui n'offre que les fonctions principales, et une interface complète. Il peut également modifier quelque peu l'habillage graphique de l'interface.
- Lorsqu'un site accueille plusieurs rédacteurs, SPIP devient un outil de travail coopératif : débats autour des articles, système de validation, travail à plusieurs sur un même article...

Paramétrage du site

Si l'interface graphique du site public et la gestion du contenu sont, dans SPIP, strictement séparées (par exemple, on ne fixe pas la couleur du fond d'écran du site public dans l'espace privé), il est cependant possible de configurer certains comportements du site dans l'espace privé :

- accepter ou refuser certains éléments du contenu des articles : ainsi on peut décider d'interdire l'utilisation des surtitre, sous-titre, descriptif, chapeau ou post-scriptum, ou la date de publication antérieure et les mots-clés ;
- configurer (ou désactiver) les forums publics ;
- indiquer si l'on publie les articles avant la date de publication qu'on leur a fixé (cette option permet par exemple de partir en vacances, le site publiant des articles pendant cette absence) ;
- désactiver le système de brèves (en effet, certains sites n'en ont pas l'usage ; les désactiver permet de simplifier l'interface pour les rédacteurs) ;
- activer ou désactiver les statistiques ;
- activer ou désactiver le moteur de recherche.




Suivi éditorial

Afin de faciliter le suivi éditorial du site, plusieurs options sont offertes :

- envoi des contributions des forums aux auteurs des articles ; lorsqu'un visiteur du site poste un message sous un article, l'auteur de cet article en est informé par mail, ce qui lui permet de suivre l'activité de son article par mail ;
- suivi de l'activité éditoriale ; si le site est le fruit d'une équipe de rédacteurs, on peut signaler automatiquement les annonces importantes de l'activité éditoriale à une adresse email (dans l'idéal, une liste de diffusion) ; ainsi, lorsqu'un article est publié ou proposé à la publication, cette liste en est informée ;
- annonce des nouveautés ; SPIP peut envoyer automatiquement, selon une fréquence fixée par les administrateurs, un courrier électronique recensant les dernières publications sur le site.

Il existe dans SPIP trois sortes d'auteurs : les visiteurs, les rédacteurs et les administrateurs.

6.3 Droits dans SPIP

-  **Le visiteur** est une personne qui peut simplement accéder aux parties restreintes du site public ainsi qu'aux forums auquel il est abonné. Il n'a aucun accès à la partie privée (écrire) du site.
-  **Le rédacteur** est une personne qui a accès à l'espace privé du site et qui ne peut que **rédiger des textes** et proposer des contenus (textes et documents multimédia) à l'évaluation. C'est le statut le plus couramment utilisé.
-  **L'administrateur** est une personne qui a accès à l'espace privé et qui a **« tous les droits » sur le site** : ajout, modification et suppression des contenus, relecture et correction des textes rédigés par les autres rédacteurs, modification l'arborescence, etc. et surtout publication sur le site public. Il est le responsable moral du contenu du site.
 - **L'administrateur restreint** est un administrateur dont les droits sont restreints à une ou plusieurs rubriques.

| Exemples de droits typiques des auteurs SPIP | | |
|--|--|--|
| Visiteur | Rédacteur | Administrateur |
| <p>Il peut :</p> <ul style="list-style-type: none"> • visiter le site public • Participer aux forums sur abonnement du site public [*] | <p>Il peut faire tout ce qu'un visiteur peut faire mais aussi :</p> <ul style="list-style-type: none"> • rédiger de nouveaux contenus : écrire un nouvel article, une nouvelle brève [*], ajouter des images et documents, référencer un site web [*], etc. • proposer de nouveaux contenus à la relecture avant publication • continuer à écrire et corriger ses articles « en cours de rédaction » ou « proposés à la publication » ; | <p>Il peut faire tout ce qu'un rédacteur peut faire mais aussi :</p> <ul style="list-style-type: none"> • Publier les contenus sur le site public : valider et publier un article, une brève, un site, etc. • Modifier les contenus déjà publiés • Associer un nouvel auteur à un article • Modifier les droits et statuts des autres auteurs • Créer de nouveaux |

| Exemples de droits typiques des auteurs SPIP | | |
|--|--|--|
| Visiteur | Rédacteur | Administrateur |
| | <ul style="list-style-type: none"> • consulter et donner son avis pour les articles des autres rédacteurs, proposés à la publication ; • modifier une brève « à valider » [*] ; • pré visualiser les articles « proposés à la publication » ; • visualiser la liste des auteurs • visualiser les mots-clés [*] Il ne peut pas : <ul style="list-style-type: none"> • Modifier un article publié, même s'il en est l'auteur • Créer un nouvelle rubrique | auteurs <ul style="list-style-type: none"> • Modifier les paramètres de configuration générale du site • Activer, désactiver et paramétrer des fonctionnalités supplémentaires |

SPIP ne propose par défaut que trois statuts différents : ceux-ci suffisent largement à la gestion de bien des sites. Cependant, il est possible d'étendre la liste de ces statuts et de diversifier les droits, notamment par l'ajout de plugins spécifiques.

6.4 Installer SPIP

Aucune configuration spéciale n'est nécessaire sur votre ordinateur personnel, SPIP se gère entièrement sur le Web. Tout ce dont vous avez besoin, c'est d'un **navigateur Web** (n'importe lequel) et d'un **accès FTP** pour installer les fichiers sur votre espace d'hébergement, que ce soit via une interface en ligne ou avec un logiciel de transfert FTP installé sur votre ordinateur.

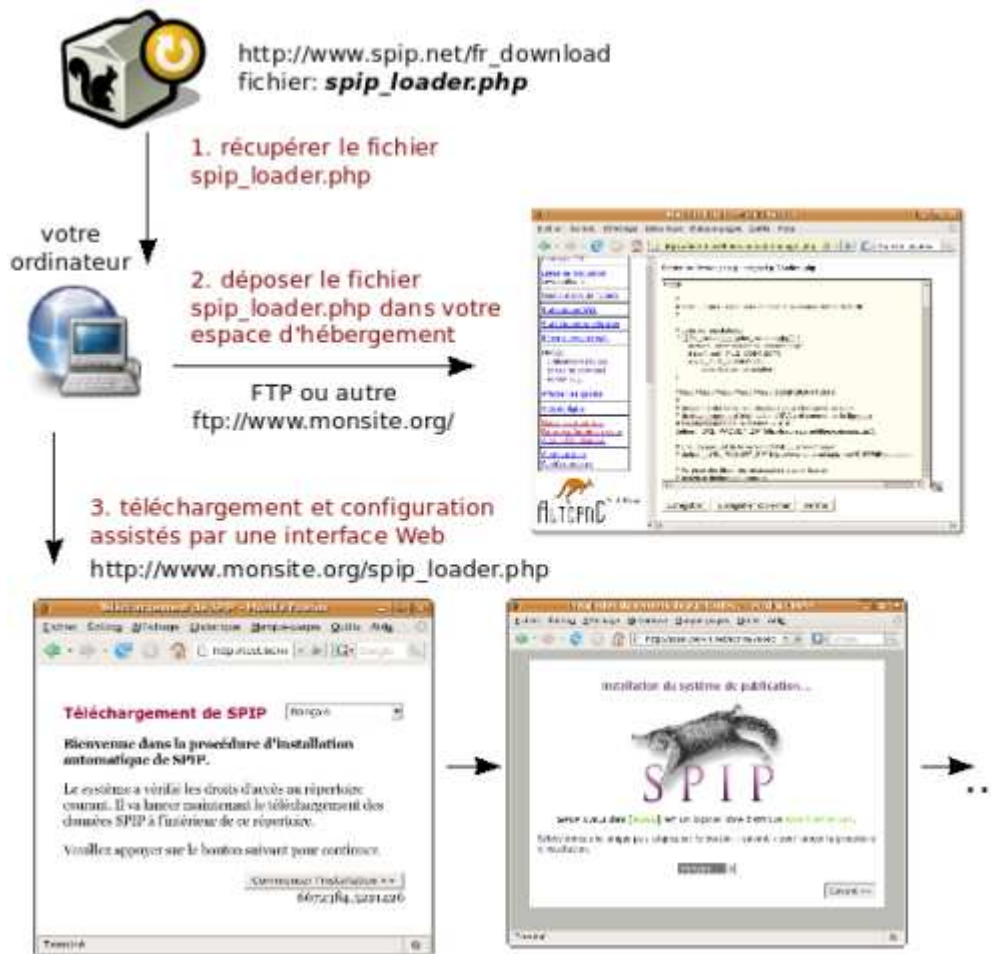
Pour installer un site SPIP, il vous faudra un espace d'hébergement web ordinaire pourvu d'une base de données MySQL et capable d'exécuter du PHP. Mais soyez rassuré : aucune connaissance de PHP ou de MySQL n'est nécessaire pour

administrer un site SPIP ! Il vous faudra juste vérifier que votre hébergeur le propose. Voir : « Hébergeurs acceptant SPIP ».

Munissez-vous de vos accès, identifiants et mots de passe FTP et MySQL. Ces éléments sont indispensables : si vous ne les connaissez pas, contactez votre hébergeur et demandez-lui de vous les rappeler. Avant l'installation de SPIP, vous devez aussi avoir une base MySQL disponible. Sur de très nombreux hébergements, il faut soit demander l'activation d'une base MySQL à l'administrateur, soit suivre une procédure automatique en ligne (dans tous les cas, l'activation de la base MySQL n'a rien à voir avec SPIP ; si vous avez des difficultés, seul votre hébergeur peut vous fournir les mots de passe nécessaire et vous expliquer comment activer votre compte MySQL). Vous devez connaître les données de votre connexion MySQL (fournies par l'hébergeur) :

- **l'adresse de la base MySQL** : par exemple *sql.free.fr*, ou *localhost*, ou vide ;
- **votre login MySQL** : souvent le même login que votre compte Web ;
- **votre password MySQL** : souvent le même que le compte Web ;
- **le nom de la base de données** : souvent le même login que votre compte Web - il est possible que le serveur vous offre la possibilité de créer vous même cette base.

La procédure la plus facile et la plus rapide est l'installation automatique avec « *spip_loader* ». Ce petit fichier va télécharger SPIP et l'installer lui-même chez votre hébergeur. *Attention* : cette procédure ne fonctionne pas sur tous les serveurs. Si elle ne fonctionne pas (vous vous en rendrez compte immédiatement), passez directement à l'autre méthode.



Récupérez le fichier `spip_loader.php` (ou copiez juste son contenu),

Déposez ce fichier dans votre espace d'hébergement (par FTP ou par tout autre moyen que vous propose votre hébergeur). Visitez votre site avec votre navigateur habituel et affichez-y le fichier que vous venez de déposer, en vous rendant à l'adresse du style :

http://www.monsite.fr/spip_loader.php et suivez les indications.

Où placer les fichiers de squelettes ?

Depuis SPIP 1.8, les squelettes sont rangés dans un dossier dédié, nommé `dist/`. Le dossier `squelettes/` accueillera vos squelettes personnalisés. Les avantages de ce rangement sont évidents : meilleure séparation du code de SPIP et de la structure du site, possibilité de changer tout un ensemble de squelettes d'un seul coup, etc.

Les squelettes par défaut : `dist/`

Depuis SPIP 1.8, les squelettes de la distribution — c'est-à-dire ceux fournis en standard à l'installation de SPIP — sont regroupés dans le répertoire `dist/`. Ces fichiers contiennent les informations sur la mise en page par défaut du site et ne doivent pas être modifiés. Vous pouvez examiner le contenu de ce répertoire et partir de ce jeu de squelettes pour adapter la mise en page à vos besoins

Toutefois, **il ne faut surtout pas modifier les fichiers du répertoire `dist/`**, sinon vous risqueriez de perdre toutes vos modifications à chaque mise à jour de SPIP ! Pour éviter cela, **faites une copie des fichiers que vous souhaitez modifier**, et placez-les dans un autre répertoire, comme indiqué ci-après.

Votre dossier squelettes/

Depuis SPIP 1.8, les squelettes personnalisés doivent être rangés dans un répertoire nommé `squelettes/` (attention au « s » final !), que vous créez à la racine de votre site SPIP. Que vous souhaitiez installer un jeu complet de squelettes (pris sur SPIP - Contrib ou ailleurs), ou apporter une légère modification aux squelettes par défaut, placez vos squelettes dans ce répertoire.

Ainsi, un utilisateur qui veut créer sa propre mise en page, développera ses propres fichiers `article.html`, `rubrique.html`, etc. dans le répertoire `squelettes/`. Notez bien qu'il n'est pas indispensable de placer un jeu de squelettes complet dans ce répertoire.

Pour afficher les pages du site, SPIP cherche les squelettes prioritairement dans le dossier `squelettes/` ; si SPIP n'y trouve pas un fichier `.html` qui lui est nécessaire, il ira chercher celui de la distribution dans le dossier `dist/`.

Ainsi, si vous n'avez placé qu'un seul fichier dans le dossier `squelettes`, par exemple `article.html`, SPIP utilisera ce squelette pour afficher les articles, et ceux de la `dist` pour toutes les autres pages du site.

Le dossier `squelettes/` est destiné à recevoir tous les fichiers nécessaires à la mise en page d'un site.

On y rangera donc :

Les squelettes, c'est-à-dire les fichiers `.html` avec du code SPIP ; les fichiers inclus dans les squelettes (ainsi que, pour les versions antérieures à SPIP 1.9, leur fichier `php3` correspondant) et les modèles (depuis SPIP 1.9) ;

les formulaires modifiés, de préférence dans un sous-répertoire `formulaires/`
 les feuilles de style CSS qui produisent l'habillage graphique ; les personnaliser permet en effet de varier, parfois spectaculairement, l'habillage d'un site sans intervenir dans les squelettes. Voir : « Mettez-y votre style ! » ;
 les images utilisées dans les squelettes ;
 le fichier `mes_fonctions.php` contenant les filtres et variables de personnalisation propres à ce jeu de squelettes ;
 les fichiers javascripts ;
 les fichiers de langue personnalisés (cf. : « Internationaliser les squelettes », méthode des fichiers de langues), de préférence dans un sous-répertoire `lang/` ;
 etc...

6.5 Des boucles et des balises

Tout ce qui suit concerne désormais le langage de description de la mise en page des *squelettes* dans SPIP ; si vous avez bien compris l'explication précédente, vous savez que nous travaillons donc dans les fichiers « .html ».

La présente documentation est volontairement technique (il s'agit ici de références techniques) ; vous pouvez préférer commencer par notre guide Pas à pas, plus didactique, et revenir ensuite ici pour une documentation plus précise.

Des boucles

La notion de base du langage de SPIP est la *boucle*.

- **La logique de la boucle**

Une base de données, classiquement, c'est une liste d'éléments : ici, une liste des articles, une liste des rubriques, une liste des auteurs, etc. Pour « fabriquer » le site, on va donc extraire de cette liste certains de ses éléments :

- à la base, on veut extraire *un seul élément* d'une liste ; par exemple, afficher l'article désiré ;
- mais il est fréquent d'extraire *plusieurs éléments* d'une liste ; par exemple, dans la page d'une rubrique, on veut afficher *tous* les articles contenus dans cette rubrique, ainsi que *toutes* les sous-rubriques contenues dans cette rubrique ;
- plus subtil : il arrive fréquemment qu'il n'y ait pas d'éléments satisfaisants à tel ou tel endroit ; SPIP doit alors pouvoir gérer l'éventualité de l'absence de ces éléments ;

par exemple, le squelette de l'affichage des rubriques demande l'affichage de toutes les sous-rubriques contenues dans une rubrique ; que faire, alors, s'il n'y a pas sous-rubriques dans cette rubrique spécifique ?

Ces trois situations sont traitées par la notion unique de *boucle*, qui permet à la fois de gérer l'affichage d'un seul élément, de plusieurs éléments successifs, ou l'absence d'éléments.

Le système de boucle permet, dans un code unique :

- d'indiquer à quel endroit du code HTML on a besoin de quel type d'élément (à tel endroit on veut récupérer la liste des articles, à tel endroit on veut inclure la liste des sous-rubriques...) ;
- de prévoir l'affichage d'un élément unique ;
- d'indiquer comment est affichée une liste de plusieurs éléments ;
- de déterminer ce qu'on affiche lorsqu'il n'y a aucun élément correspondant.

Analogie avec la programmation en PHP/mysql

Ceux qui ont déjà programmé des requêtes mysql en PHP savent que le traitement se déroule en deux temps :

- la construction de la syntaxe de la requête (qui consiste à dire « je veux récupérer la liste des articles contenus dans telle rubrique... ») ;
- l'analyse et l'affichage des résultats au travers d'une boucle.

Ce sont ces deux événements qui sont gérés, dans SPIP, au travers des boucles.

Les balises SPIP

Grâce aux boucles, on a donc récupéré des éléments uniques ou des listes d'éléments : par exemple une liste d'articles ou une liste de rubriques...

Cependant, chaque élément de telles listes est composé de plusieurs éléments précis : par exemple un article se compose d'un titre, d'un surtitre, d'un sous-titre, d'un texte d'introduction (chapeau), d'un texte principal, d'un post-scriptum, etc. Il existe ainsi des balises spécifiques à SPIP, permettant d'indiquer précisément à quel endroit on affiche des éléments : « placer le titre ici », « placer le texte ici »...

Les balises à l'intérieur des boucles

Voici, au travers d'un cas classique, le principe de fonctionnement général d'une boucle accompagnée de ses balises (attention, ça n'est pas du langage SPIP, c'est une description logique) :

BOUCLE : afficher la liste des articles de cette rubrique

afficher ici le titre de l'article

afficher le sous-titre

afficher le texte

Fin de la BOUCLE

Cette boucle, analysée par SPIP, peut donner trois résultats différents.

- Il n'y a aucun article dans cette rubrique.

Dans ce cas, bien évidemment, aucun des éléments « afficher ici... (titre, sous-titre...) » n'est utilisé. En revanche, si on l'a prévu, on peut afficher un message du genre « Il n'y a pas d'article ».

- Il y a un seul article dans cette rubrique.

Dans ce cas, très simplement, la page HTML est construite sur le modèle de la boucle :

Titre de l'article

Sous-titre

Texte de l'article

- Il y a plusieurs articles dans cette rubrique.

La description de la mise en page (« placer ici... ») va alors être calculée successivement pour chacun des articles. Ce qui donne simplement :

Titre de l'article 1

Sous-titre de l'article 1

Texte de l'article 1

Titre de l'article 2

Sous-titre de l'article 2

Texte de l'article 2

...

Titre du dernier article

Sous-titre du dernier article

Texte du dernier article

La suite de ce guide de référence se construira donc de la manière suivante :

- syntaxe générale des boucles ;
- syntaxe générale des balises de SPIP ;
- et, ensuite, une page spécifique à chaque type de boucles, indiquant quelles balises on peut y utiliser.

La syntaxe des boucles

Syntaxe de base

La syntaxe simplifiée d'une boucle est la suivante :

```
<BOUCLE $n$ (TYPE){critère1}{critère2}...{critèrex}>
```

* Code HTML + balises SPIP

```
</BOUCLE $n$ >
```

On a vu, dans l'explication sur les boucles et les balises, que le « Code HTML + balises SPIP » se répétait autant de fois que la boucle obtenait d'éléments tirés de la base de données (c'est-à-dire une fois, plusieurs fois, ou zéro fois).

La ligne importante, ici, est :

```
<BOUCLE $n$ (TYPE){critère1}{critère2}...{critèrex}>
```

- **L'élément** `BOUCLE` est l'ordre indiquant qu'il s'agit d'une boucle SPIP ; on ne peut donc pas le modifier ; dit autrement, toutes les boucles de SPIP commencent par l'instruction `BOUCLE`.

- **L'élément** n est le nom ou le numéro de la boucle, librement choisi par le webmestre, pour chaque boucle qu'il utilise. On verra plus loin qu'il est possible (c'est même tout l'intérêt de la manœuvre) d'utiliser plusieurs boucles dans un même squelette : les nommer est donc indispensable pour les identifier.

Si vous décidez de numéroter vos boucles, la syntaxe devient par exemple (pour la boucle 5) :

```
<BOUCLE5...> ... </BOUCLE5>
```

Si vous décidez de donner un nom à vos boucles (c'est généralement plus pratique, votre code est plus lisible), il faut impérativement faire précéder ce nom par le symbole « _ » (que l'on appelle habituellement *underscore*). Par exemple :

```
<BOUCLE_sousrubriques...> ... </BOUCLE_sousrubriques>
```

- **L'élément** `(TYPE)` est primordial : il indique quel type d'éléments on veut récupérer.

La syntaxe est importante : le TYPE est indiqué entre parenthèses (sans espaces), en majuscules, et ce TYPE doit correspondre obligatoirement à l'un des types prévus dans SPIP (qu'on trouvera dans la présente documentation) : ARTICLES, RUBRIQUES, AUTEURS, BREVES, etc.

Pour l'exemple précédent, on aurait donc :

```
<BOUCLE_sousrubriques(RUBRIQUES)...>
```

```
...
```

```
</BOUCLE_sousrubriques>
```

- **Les critères** `{critère1}{critère2}...{critèrex}` indiquent à la fois selon quels critères on veut sélectionner les éléments de la base de données (afficher les sous-rubriques

incluses dans cette rubrique, afficher les autres rubriques installées au même niveau hiérarchique que la présente rubrique...), et la façon dont on va classer ou sélectionner les éléments (classer les articles selon leur date, selon leur titre... afficher uniquement les 3 premiers articles, afficher la moitié des articles...). Comme on peut combiner les critères, on peut très aisément fabriquer des requêtes très puissantes, du genre « afficher la liste des 5 articles les plus récents écrits par cet auteur ». Les critères sont entre accolades ; ils peuvent être séparés les uns des autres par un espace. Exemple :

```
<BOUCLE_meme_auteur(ARTICLES)      {id_auteur}      {par      date}{inverse}      {0,5}>
...
</BOUCLE_meme_auteur>
```

Les différents critères et leur syntaxe seront explicités par la suite, pour chaque type de boucle (certains critères fonctionnent pour tous les types de boucles, d'autres sont spécifiques à certaines boucles).

Syntaxe complète

Le syntaxe indiquée précédemment peut être complétée par des éléments conditionnels. En effet, la boucle précédente affiche successivement les éléments contenus à l'intérieur de la boucle. SPIP permet de plus d'indiquer ce qu'on affiche avant et après la boucle au cas où elle contient un ou plusieurs résultats, et ce qu'on affiche s'il n'y a aucun élément.

Cela donne :

<Bn>

* Code HTML optionnel avant

<BOUCLE_n(TYPE){critère1}{critère2}...{critèrex}>

* Code HTML + balises SPIP

</BOUCLE_n>

* Code HTML optionnel après

</Bn>

* Code HTML alternatif

</Bn>

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

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

Le code *alternatif* (terminé par `</Bn>`) est affiché à la place de la boucle (et donc également à la place des codes optionnels avant et après) si la boucle n'a trouvé aucune réponse.

Par exemple, le code :

```
<B1>
  Cette rubrique contient les éléments suivants:
  <ul>
    <BOUCLE1(ARTICLES){id_rubrique}>
    <li>#TITRE</li>
  </BOUCLE1>
  </ul>
</B1>
  Cette rubrique ne contient pas d'article.
</B1>
```

donne les résultats suivants :

- **s'il y a un seul article :**

```
Cette rubrique contient les éléments suivants:
<ul>
  <li>Titre de l'article</li>
</ul>
```

- **s'il y a plusieurs articles :**

```
Cette rubrique contient les éléments suivants:
<ul>
  <li>Titre de l'article 1</li>
  <li>Titre de l'article 2</li>
  ...
  <li>Titre du dernier article</li>
</ul>
```

- **s'il n'y a aucun article :**

```
Cette rubrique ne contient pas d'article.
```

Historique : Jusqu'à [SPIP 1.7.2], La manière dont SPIP interprétait les boucles interdisait de mettre une boucle entre `<Bn>` et `<BOUCLEn>`. Par contre, il restait possible de mettre des boucles supplémentaires dans les parties optionnelles situées après la

définition `<BOUCLEn...>`. Si vous deviez vraiment installer une boucle dans la partie optionnelle *avant*, il fallait passer par une commande `<INCLURE () >`

La boucle ARTICLES

Une boucle d'articles se code en plaçant entre parenthèses `ARTICLES` (avec un « s ») :

```
<BOUCLEn(ARTICLES){critères...}>
```

Les éléments contenus dans une telle boucle sont des articles.

Les critères de sélection

On utilisera l'un ou autre des critères suivants pour indiquer comment on sélectionne les éléments.

- `{tout}` : les articles sont sélectionnés dans l'intégralité du site (dans toutes les rubriques). Utile notamment pour afficher les articles les plus récents (dans l'intégralité du site) sur la page d'accueil. [En réalité, le critère « tout » n'est pas traité de manière informatique : c'est un aide-mémoire pour le webmestre ; on obtient le même résultat en n'indiquant aucun des critères suivants.]
- `{id_article}` sélectionne l'article dont l'identifiant est `id_article`. Comme l'identifiant de chaque article est unique, ce critère ne retourne qu'une ou zéro réponse.
- `{id_rubrique}` sélectionne les articles contenus dans la rubrique dont l'identifiant est `id_rubrique`.
- `{id_secteur}` sélectionne les articles dans ce secteur (un secteur est une rubrique qui ne dépend d'aucune autre rubrique, c'est-à-dire située à la racine du site).
- `{branche}` (depuis [SPIP 1.4](#)) sélectionne l'ensemble des articles de la rubrique ET de ses sous-rubriques. (C'est une sorte d'extension du critère `{id_secteur}`). Toutefois, à l'inverse de `{id_secteur=2}`, il n'est pas possible d'appeler directement une *branche* en faisant par exemple `{branche=2}` : techniquement parlant, il faut que la rubrique en question figure dans le contexte courant. Ce critère est à utiliser avec parcimonie : si votre site est bien structuré, vous ne devriez pas en avoir besoin, sauf dans des cas très particuliers.)
- `{id_auteur}` sélectionne les articles correspondant à cet identifiant d'auteur (utile pour indiquer la liste des articles écrits par un auteur).

- `{id_mot}` sélectionne les articles correspondant à cet identifiant de mot-clé (utile pour indiquer la liste des articles traitant d'un sujet donné).
- `{titre_mot=xxxx}`, ou `{type_mot=yyyy}` (depuis [SPIP 1.3](#)) sélectionne respectivement les articles liés au mot-clé dont le nom est « xxxx », ou liés à des mots-clés du groupe de mots-clés « yyyy ». Si l'on donne plusieurs plusieurs critères `{titre_mot=xxxx}` (ou plusieurs `{type_mot=yyyy}`), on sélectionnera ceux qui auront tous ces mots à la fois (nouveau de [SPIP 1.9](#)).
- `{id_groupe=zzzz}` (depuis [SPIP 1.4](#)) permet de sélectionner les articles liés à un groupe de mots-clés ; principe identique au `{type_mot}` précédent, mais puisque l'on travaille avec un identifiant (numéro du groupe), la syntaxe sera plus « propre ». [Nota : Ce critère n'est pas (en l'état actuel du développement de SPIP) cumulable avec le précédent `{type_mot=yyyy}`]
- `{lang}` (depuis [SPIP 1.7.1](#)) sélectionne les articles de la langue demandée dans l'adresse de la page.
- `{traduction}` (depuis [SPIP 1.7.1](#)) sélectionne les traductions de l'article courant en différentes langues.
- `{origine_traduction}` (depuis [SPIP 1.7.1](#)) sélectionne l'article de référence dont l'article courant est une traduction.
- Les critères `{date}` (ou `{date=...}` ou `{date==...}`) permettent de sélectionner un article en fonction de la date passée dans l'URL (depuis [SPIP 1.7.2](#)).
- `{recherche}` sélectionne les articles correspondant aux mots indiqués dans l'interface de recherche (moteur de recherche incorporé à SPIP). Voir la page consacrée au moteur de recherche.

Le statut de l'article

Comme toutes les boucles de SPIP, une boucle `ARTICLES` ne retourne que des articles *publiés* ; dans le cas où le site est réglé de manière à ne pas publier les articles « postdatés », un autre test est fait sur la date de l'article. Jusqu'à [SPIP 1.8.2](#) il n'existait aucun moyen de débrayer ce système et d'afficher les articles « en cours de rédaction », « proposés à la publication » ou « refusés ». C'est désormais possible grâce au critère `{statut}` :

`{statut=prop|prepa|publie|refuse|poubelle}` (depuis SPIP 1.8.2) sélectionne les articles en fonction de leur statut de publication :

`{statut=prepa}` sélectionne les articles en cours de rédaction dans l'espace privé ;

`{statut=prop}` sélectionne les articles proposés à la publication ;

`{statut=publie}` sélectionne les articles publiés sur le site, y compris les articles « postdatés » ;

`{statut=refuse}` sélectionne les articles qui ont été refusés à la publication ;

`{statut=poubelle}` sélectionne les articles qui ont été mis à la poubelle.

Les critères d'affichage

Une fois fixé l'un des critères ci-dessus, on pourra ajouter les critères suivants pour restreindre le nombre d'éléments affichés.

Les critères communs à toutes les boucles s'appliquent évidemment.

Les balises de cette boucle

Les balises tirées de la base de données

Les balises suivantes correspondent aux éléments directement tirés de la base de données. Vous pouvez les utiliser également en tant que critère de classement (par exemple : `{par date}` ou `{par titre}`).

- `#ID_ARTICLE` affiche l'identifiant unique de l'article. Utile pour fabriquer des liens hypertextes non prévus (par exemple vers une page « Afficher au format impression »).
- `#SURTITRE` affiche le surtitre.
- `#TITRE` affiche le titre de l'article.
- `#SOUSTITRE` affiche le sous-titré.
- `#DESCRIPTIF` affiche le descriptif.
- `#CHAPO` affiche le texte d'introduction (chapeau).
- `#TEXTE` affiche le texte principal de l'article.

- `#PS` affiche le post-scriptum.
- Les balises de dates : `#DATE`, `#DATE_REDAC`, `#DATE_MODIF` sont explicitées dans la documentation sur « La gestion des dates ».
- `#ID_RUBRIQUE` affiche l'identifiant de la rubrique dont dépend l'article.
- `#ID_SECTEUR` affiche l'identifiant du secteur dont dépend l'article (le secteur étant la rubrique parente située à la racine du site).
- `#NOM_SITE` et `#URL_SITE` affichent le nom et l'url du « lien hypertexte » de l'article (si vous avez activé cette option).
- `#VISITES` affiche le nombre total de visites sur cet article.
- `#POPULARITE` affiche le pourcentage de popularité de cet article ; voir la documentation « La « popularité » des articles ».
- `#LANG` affiche la langue de cet article.

Les balises calculées par SPIP

Les éléments suivants sont calculés par SPIP (Ils ne peuvent pas être utilisés comme critère de classement).

- `#URL_ARTICLE` affiche l'URL de la page de l'article.
- `#NOTES` affiche les notes de bas de page (calculées à partir de l'analyse du texte).
- `#INTRODUCTION` (depuis SPIP 1.4) affiche le descriptif de l'article, sinon affiche les 600 premiers caractères du début de l'article (chapeau puis texte). Dans les versions antérieures à SPIP 1.3, ce sont systématiquement les premiers caractères de l'article (chapeau puis texte) qui sont pris en compte (le descriptif n'est pas utilisé).
- `#LESAUTEURS` affiche les auteurs de cet article, avec lien vers leur propre page publique (afin de pouvoir directement leur écrire ou de consulter la liste des articles qu'ils ont publié). Cela évite de créer une boucle AUTEURS pour obtenir le même résultat. Dans les versions antérieures à SPIP 1.9, cette balise affiche les auteurs de l'article avec lien vers leur adresse e-mail.
- `#PETITION` affiche le texte de la pétition si elle existe. Si elle existe mais que le texte est vide, retourne un espace (une chaîne non vide sans incidence dans une page html).
- `#FORMULAIRE_SIGNATURE` fabrique et affiche le formulaire permettant de signer la pétition associée à cet article.

- `#FORMULAIRE_FORUM` fabrique et affiche le formulaire permettant de poster un message répondant à cet article. Pour en savoir plus, voir aussi « Les formulaires ».
- `#PARAMETRES_FORUM` fabrique et affiche la liste des variables exploitées par le formulaire permettant de répondre à cet article. Par exemple :

Depuis SPIP 1.8.2 on peut lui passer un paramètre spécifiant l'adresse de retour après avoir posté le message. Par exemple :

```
<a href="spip.php?page=forum&(#PARAMETRES_FORUM{#SELF})">Répondre à cet article</a>
```

renverra le visiteur sur la page actuelle une fois que le message a été validé.

Historique : Dans les versions antérieures à SPIP 1.9 il aurait fallu écrire `forum.php3?` et non `spip.php?page=forum&`

De façon générale jusqu'à SPIP 1.9, les urls des pages générées par SPIP étaient de la forme `http://monsite.net/xxx.php3` et non pas `http://monsite.net/spip.php?page=xxx`.

Les logos

- `#LOGO_ARTICLE` affiche le logo de l'article, éventuellement avec la gestion du survol.
- `#LOGO_RUBRIQUE` affiche le logo de la rubrique de l'article.
- `#LOGO_ARTICLE_RUBRIQUE` affiche le logo de l'article, éventuellement remplacé par le logo de la rubrique s'il n'existe pas de logo spécifique à l'article.

Les logos s'installent de la manière suivante : `[(#LOGO_ARTICLE|alignement|adresse)]`

L'alignement peut être `left` ou `right`. L'adresse est l'URL de destination du lien de ce logo (par exemple `#URL_ARTICLE`). Si l'on n'indique pas d'adresse, le bouton n'est pas cliquable.

Si l'on veut récupérer directement le nom du fichier du logo (alors que les balises précédentes fabriquent le code HTML complet pour insérer l'image dans la page), par exemple pour afficher une image en fond de tableau, on utilisera le filtre

`|fichier` comme suit : `[(#LOGO_ARTICLE|fichier)]`

Par ailleurs deux balises permettent de récupérer un seul des deux logos :

- `#LOGO_ARTICLE_NORMAL` affiche le logo sans survol ;
- `#LOGO_ARTICLE_SURVOL` affiche le logo de survol.

Sources

| nom | Adresse |
|----------------------|--|
| Base de connaissance | http://fr.wikipedia.org/ http://www.commentcamarche.net/ |



Informatique
et solidarité

| | |
|--------------------|---|
| | http://www.futura-sciences.com/fr/ http://lea-linu.org |
| News | http://www.clubic.com/ http://www.zdnet.fr/ |
| Site internet | http://www.allhtml.com/ http://www.nexen.net/ |
| Javascript | http://www.toutjavascript.com/main/index.php3 http://www.editeurjavascript.com/ http://www.easy-script.com/ |
| Gestion de contenu | http://www.cmsmatrix.org/matrix http://www.spip.net/fr |
| Projets | http://www.freashmeat.fr http://www.sourceforge.fr |
| Logiciels gratuit | http://www.framasoft.fr |