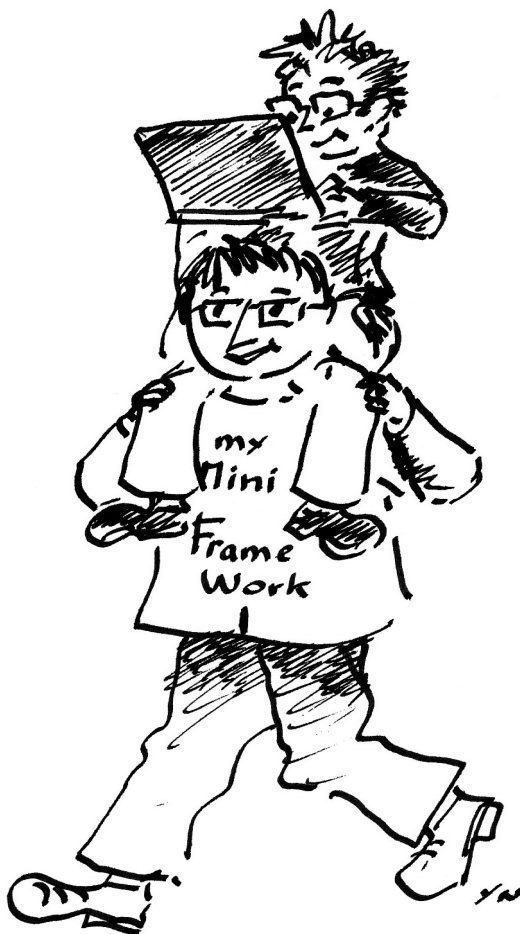


Pierre
Contri

myMiniFramework for meta web site



Mini Framework pour site web basé sur PHP

Août 2012

MYMINIFRAMEWORK FOR META WEB SITE

Pierre Contri

myMiniFramework for meta web site

© Pierre Contri, 2012

Toute reproduction ou publication, même partielle, de cet ouvrage est interdite sans l'autorisation préalable de l'auteur.

A ma famille

Sommaire

- Introduction.....9
 - Introduction.....10
 - Introduction philosophique.....11
 - Remerciements.....12
- Présentation du Framework.....13
 - Cahier des charges.....14
 - Objectif premier du mini Framework.....15
 - Pourquoi créer un outil supplémentaire ?.....15
 - Objectif et public visé.....16
 - Qu'entend-on par « procédural avec philosophie objet » ?.....17
- Présentation du Modèle.....19
 - Point de départ.....20
 - Gestion des sous-sites dans le site web.....21
 - Organisation du site et des sous-sites.....22
 - Le site.....22
 - Les sous-sites.....23
 - Les catégories.....23
 - Hiérarchie physique du Framework.....24
 - Aperçu du contenu physique du Framework.....24
 - Explication principale des différents répertoires et fichiers :.....26
 - Le système de Base de Données du Framework.....27
 - La gestion des scripts clients JavaScript.....28
 - MusicManagement.js.....28
 - PictureManagement.js.....29
 - SiteManagement.js.....32
 - WaitingManagement.js.....33
- Créer un site (ou sous-site) web.....35
 - Le fichier patron de création d'un sous-site.....36
 - Étapes de création d'un site web avec le Framework.....37
 - Créer un menu pour un sous-site.....38
 - La balise « entry ».....39
 - La balise « menu ».....39
 - Utiliser la base de données pour gérer les objets multimédias.....40

| | |
|--|-----|
| Les diagrammes UML simplifiés du Framework..... | 43 |
| Le diagramme de classe global..... | 44 |
| Le diagramme de classe global (suite)..... | 45 |
| Les classes Documents..... | 46 |
| Le diagramme de classe de la fabrique de documents..... | 48 |
| Le diagramme de classe des pages Web..... | 49 |
| Le diagramme de classe du menu..... | 50 |
| Les fonctions principales du Framework..... | 51 |
| Fonctions basiques du Framework..... | 52 |
| Fonction liée à la base de données du Framework..... | 52 |
| Fonction liée à la structure du serveur web physique..... | 53 |
| Fonctions liées aux répertoires et fichiers de l'ordinateur..... | 54 |
| Fonctions avancées du Framework..... | 55 |
| Description..... | 55 |
| Les deux objets à utiliser : « DocumentsList » et « Document »..... | 56 |
| La fabrique de documents (« FactoryDocument »)..... | 60 |
| Utilisation des fonctions du Framework..... | 62 |
| Utilisation de l'affichage d'un tableau de photos dans une page web par l'exemple..... | 62 |
| Utilisation de l'affichage d'un tableau de morceaux audio dans une page web par l'exemple..... | 64 |
| Utilisation mélangée du Framework et d'une partie HTML..... | 66 |
| Exemple de « Hello World » avec ce Framework..... | 69 |
| Création du fichier de données (« helloworld.php »)..... | 70 |
| Création du fichier de style (« helloworld.css »)..... | 72 |
| Annexes..... | 75 |
| Code source de mon propre serveur Web basé sur PHP..... | 76 |
| Code source des fichiers Javascript..... | 83 |
| Gestion des fichiers « audio »..... | 83 |
| Gestion des fichiers « image »..... | 84 |
| Gestion du site..... | 93 |
| Gestion de l'attente entre l'appel client et le serveur..... | 97 |
| Code source des principales classes en PHP..... | 99 |
| Gestion des constantes avant l'entrée des objets..... | 99 |
| La classe abstraite « Documents »..... | 100 |

| | |
|------------------------------------|-----|
| La classe « PictureDocument »..... | 102 |
| La classe « MusicDocument »..... | 105 |
| La classe « TextDocument »..... | 105 |
| La classe « ToolDocument »..... | 106 |
| La classe « UnknownDocument »..... | 106 |
| La classe « DocumentsList »..... | 107 |
| La classe « FactoryDocument »..... | 110 |
| La classe « WebPage »..... | 112 |
| La classe « WebMasterPage »..... | 113 |
| Bibliographie..... | 115 |
| Sites Web de référence..... | 116 |
| Site Web de l'auteur..... | 116 |
| Contact..... | 116 |

Introduction

Contenu de ce chapitre

| | |
|---------------------------------|----|
| Introduction..... | 9 |
| Introduction..... | 10 |
| Introduction philosophique..... | 11 |
| Remerciements..... | 12 |

Introduction

Bien que tenté par l'obtention d'un doctorat (peut compatible avec ma vie familiale), je me suis arrêté à des études d'ingénieur en informatique industrielle et génie logiciel.

J'ai décidé de publier, de manière physique, un travail que j'ai accompli en dix années de travail en pointillé. Ce dernier est très loin de la perfection que j'aimerais tant atteindre, mais a l'avantage d'être petit, relativement léger (pour de l'objet) et facile à utiliser.

Je travaillerai sur ce projet encore longtemps, pour le rendre de plus en plus transparent, jusqu'à son extinction avec l'utilisation d'autres Frameworks bien meilleurs.

Je voudrais aussi montrer que je suis loin d'être le meilleur informaticien du monde, mais pas non plus le plus mauvais ; et puis enfin montrer mon objectif d'apprendre toujours et d'être en permanence à la recherche du savoir. Admettre de manquer de savoir permet d'en acquérir. Si l'on pense tout connaître, à quoi cela sert-il de continuer à vivre ?

Au travers de la présentation de ce Framework fiable, je voudrais montrer mes capacités de travail et proposer mes services informatiques à une société en accord avec mes envies : la quête du savoir, la volonté de se dépasser (à son rythme) et le respect de la personne humaine dans un objectif de carrière à long terme.

Introduction philosophique

Un grand respect pour Microsoft® (DotNet), Unix (C ANSI), Linux, PHP et Python qui sont mes modèles intellectuels.

Passionné par différentes façons de penser, je suis de temps en temps déçu par l'avancée trop rapide de l'informatique. En effet, prendre le temps de réaliser de bons outils serait mon rêve, même si ce ne devait pas être notre génération qui en profiterait.

Un inventeur ne peut jamais profiter pleinement de son invention, puisqu'elle est améliorée pendant des décennies par d'autres personnes. C'est grâce à cela que notre société est si moderne. Et pourquoi fait-on autrement en informatique ?

Lorsqu'une idée est émise, cette dernière doit être réalisée dans les jours qui viennent. Ce qui nuit gravement à sa qualité. Pour les industriels informatiques, la qualité n'est pas forcément le but recherché. Les idées et programmes étant éphémères, il faut seulement créer les programmes le plus rapidement possible, les rendre attirants, beaux et attrayants. Bref, un marketing de haute qualité, avec un code médiocre. La vie que je mène, ainsi que mon éducation m'ont appris le contraire. Cependant, mille fois hélas, j'ai dû admettre à mes dépends qu'il fallait se ranger sur les autres pour se faire connaître. La preuve en est ce mini écrit.

Le Framework que je propose existe depuis dix années. De bonne qualité au niveau robustesse, il était écrit en procédural (mais avec philosophie objet). Je l'ai passé en objet en juillet 2012 après mille hésitations. J'ai fait cela pour me faire connaître, publier cet écrit et montrer ma capacité à travailler en objet.

Je me suis assis sur mes principes de perfection, puisque ce dernier n'est plus compatible entièrement XHTML 1.0 Strict car je l'ai passé HTML 5 en même temps qu'objet.

Remerciements

Je remercie toutes les personnes qui ont pu m'apprendre l'informatique, ainsi que toutes celles qui m'ont fait progresser intellectuellement dans ma façon de penser. Plusieurs amis informaticiens m'ont permis d'en arriver là. Je me permets de n'en citer que quelques uns. Pardon aux autres que je n'oublie pas pour autant. Mes mentors : Laurent S., Christian A., Jean M., Christophe B.

Je remercie aussi toutes les personnes suffisantes qui connaissent absolument tout sur tout en informatique ; celles aussi qui se sont engouffrées dans Java (par exemple) croyant tout connaître et se proclament « spécialistes nouvelles technologies » sans que je n'aurais pas pu avoir l'exemple à ne pas suivre en informatique : l'auto-suffisance.

Merci à mes parents sans qui le projet n'aurait pas abouti. Tout le monde sait que le mot « finalize » en informatique est automatisé. Pour cet écrit, j'ai demandé à mes parents d'implémenter ce travail.

Merci à mon parrain pour le croquis réalisé en couverture ; ce qui donne au Framework sa touche et montre que la valeur ajoutée de ce dernier est bel et bien son apparence.

Je remercie enfin ma famille qui m'accompagne dans la compréhension de ma démarche de recherche de savoir.

Présentation du Framework

Contenu de ce chapitre

| | |
|---|----|
| Présentation du Framework..... | 13 |
| Cahier des charges..... | 14 |
| Objectif premier du mini Framework..... | 15 |
| Pourquoi créer un outil supplémentaire ?..... | 15 |
| Objectif et public visé..... | 16 |
| Qu'entend-on par « procédural avec philosophie objet » ?..... | 17 |

Cahier des charges

Créé en 2002 en langage procédural, ce Framework a vu le jour car son cahier des charges ne correspondait à aucun produit proposé à cette date.

Voici les principales caractéristiques demandées :

- permettre de gérer plusieurs sous-sites web dans un seul
- ne pas utiliser la base de données MySQL de l'hébergeur non sécurisée (à l'époque)
- avoir tous les fichiers de base de données en fichiers plats pour être indépendant des plate-formes, des outils logiciels et pouvoir les modifier avec n'importe quel outil informatique open source
- être gratuit
- être facile d'emploi
- ne gérer que le minimum : affichage simple
- être compatible aux normes HTML 3, puis HTML 4.0, puis XHTML 1.0 Strict, pour arriver à HTML 5
- pouvoir copier simplement le Framework et le coller à la racine d'un serveur web sans avoir un millier de dossiers et fichiers dans cette dernière
- séparer au maximum chaque élément pour les réutiliser et rendre autonome chaque bout de code
- éviter le doublon de code
- essayer d'avoir un code épuré, malgré la complexité d'écriture suite aux versions assez anciennes de l'hébergeur
- une partie scripts serveur (PHP), une partie scripts client (Javascript)
- nécessité d'un Framework compatible PHP 4 pour des raisons historiques d'hébergement ; besoin d'un Framework procédural

Nota Bene : A l'heure actuelle, le Framework est à 98% migré en objet. Seul les outils non utiles au développement rapide sont encore en procédural. Ce Framework, en constante évolution, peut contenir de nouvelles fonctions non décrites dans ce document. Seule la philosophie est importante.

Objectif premier du mini Framework

Le Framework doit permettre de créer rapidement un ou plusieurs sites web à l'intérieur d'un site web (enveloppe) sur des systèmes extrêmement légers et restreints en terme d'architecture physique, logiciel, etc ...

Pourquoi créer un outil supplémentaire ?

A sa création, aucun Framework existant n'était ni libre ni gratuit. Beaucoup d'éditeurs de Frameworks ne prenaient même pas la peine de respecter un minimum les règles proposées et conseillées par la W3C.

Ma volonté est de respecter les normes XHMLT 1.0 Strict et CSS 3.

J'ai eu aussi envie de montrer la possibilité et la volonté de réaliser un travail informatique.

La dernière raison d'avoir créé ce Framework est que je n'avais pas internet chez moi. J'étais donc contraint d'utiliser les livres PHP, HTML, CSS et ne pouvais pas disposer des palettes extraordinaires du Web.

Objectif et public visé

Sites web informatifs. Le moteur n'est pas conçu pour échanger de l'information, mais seulement en déposer par le propriétaire du site.

Le site se doit d'être assez simple pour permettre à un utilisateur éclairé de réaliser ses pages web.

Cela peut aussi convenir pour un développeur n'ayant aucune envie de s'intéresser au code et voulant aller à l'essentiel : le contenu.

Créer plusieurs sites web à l'intérieur d'un site pour obtenir un meta site web.

Ce Framework m'a aussi permis de réaliser des sites web pour des amis en un minimum de temps. Pour ajouter une page à un site ou sous-site web, il ne faut que 10 minutes avec publication sur serveur.

La partie la plus longue reste la génération de la présentation CSS, qui n'est pas mon fort. Pour cela, l'utilisateur ou développeur doit simplement modifier les fichiers CSS fournis. Ce n'est pas la technique qui est problématique à ce niveau, c'est la qualité du graphisme final ; partie dont je me désengage.

Qu'entend-on par « procédural avec philosophie objet » ?

Le langage C 1.0 m'a permis de comprendre que penser objet peut être indépendant du langage, même si ce dernier n'est pas prévu à cet effet.

Ce magnifique langage C ne connaissait même pas le mot objet à sa sortie. Pourtant, grâce à ses structures (« typedef struct »), il est tout à fait possible d'écrire un programme objet en C 1.0. Pour cela, on utilisera un mélange de structures et de pointeurs de fonctions dans ces dernières afin de transcrire un objet.

Cela est très intéressant. En effet, cumulées à la technique des tableaux en langages évolués (PHP 4, ...) il est possible de définir ces structures (telles qu'on les conçoit en C) dans un tableau et de travailler sur des pseudo-objets en masse.

Un exemple sur un objet simplifié et schématisé pour l'occasion : le point

```
Philosophie d'écriture en C
typedef struct
{
    int x,
    int y,
    (void*)(affiche)(void *) ;
} Point ;

Philosophie d'écriture en PHP 4
$monPoint = array() ;
$monPoint['x'] = 0 ;
$monPoint['y'] = 0 ;
$monPoint['fonctionAffiche'] = affiche ;
Appel de la fonction affiche en PHP 4
$monPoint['fonctionAffiche']() ;
```

Présentation du Modèle

Contenu de ce chapitre

| | |
|--|----|
| Présentation du Modèle..... | 19 |
| Point de départ..... | 20 |
| Gestion des sous-sites dans le site web..... | 21 |
| Organisation du site et des sous-sites..... | 22 |
| Le site..... | 22 |
| Les sous-sites..... | 23 |
| Les catégories..... | 23 |
| Hiérarchie physique du Framework..... | 24 |
| Aperçu du contenu physique du Framework..... | 24 |
| Explication principale des différents répertoires et fichiers :..... | 26 |
| Le système de Base de Données du Framework..... | 27 |
| La gestion des scripts clients JavaScript..... | 28 |
| MusicManagement.js..... | 28 |
| PictureManagement.js..... | 29 |
| SiteManagement.js..... | 32 |
| WaitingManagement.js..... | 33 |

Point de départ

A la création du modèle, j'ai défini 5 couches en dérivant le modèle MVC en MMVCC. Cela m'a permis de séparer énormément les différentes données. Le modèle 5 couches dérivé était fiable et fonctionnait bien. Seulement, cela ressemblait plus à une usine à gaz qu'à un outil d'aide.

Pour rationaliser le travail, j'ai transformé ce dernier en un modèle MVC dérivé. Le vrai modèle MVC est bien meilleur que celui utilisé, mais il y a toujours la théorie et le contexte final ; c'est pour cela que je l'ai dérivé et adapté.

Ayant beaucoup travaillé avec DotNet et PHP, je me suis inspiré des philosophies et des modèles de base de ces derniers.

Je n'ai malheureusement jamais eu la chance de pouvoir travailler professionnellement avec ZEND Framework, et suis conscient que ce dernier n'est même pas comparable tant il est excellent.

Gestion des sous-sites dans le site web

Ce Framework se base sur un site qui contient un ou plusieurs sous-sites.

Si plusieurs sous-sites sont créés, il faut, dans ces conditions, créer un index de choix de sites. Le Framework propose cet index qu'il nomme « sitechoosing.php ». Il proposera, dans un premier temps, le sous-site auquel nous souhaitons accéder.

Organisation du site et des sous-sites

Le site

A la racine du site web, un fichier index (le contrôleur du modèle MVC) gère le site de manière technique.

Dans cette racine, un dossier « data » contient le fichier nommé précédemment « sitechoosing.php », ainsi qu'un répertoire par sous-site. Le nom du répertoire correspond au nom « informatique » du sous-site.

Les autres fichiers contenus dans le répertoire « data » correspondent à des fichiers de blocage interceptant toute tentative de connexion manuelle à un dossier contenu dans le répertoire (« index.php »), ainsi que des fichiers de gestion globale de la navigation par menu (« global_nav.php »).

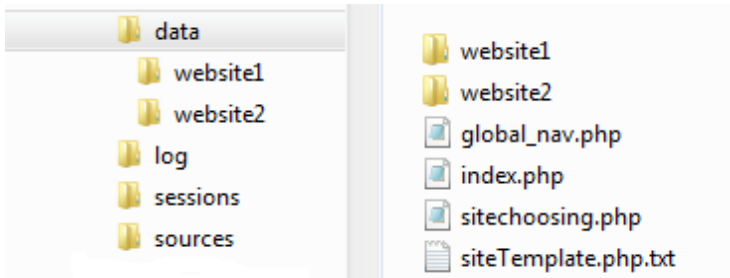


Illustration 1: Contenu du répertoire data (site)

Les sous-sites

Dans chaque répertoire de sous-sites, se trouvent :

- un fichier « index.php »
- un fichier contenant l'icône du sous-site en format « jpg »
- un fichier (ou plusieurs suivant le multi-langage) contenant les données du sous-site (exemple « monSousSite.php »)

Ces fichiers sont essentiels pour assurer le bon fonctionnement du site web.

Vous pourrez ensuite y ajouter des dossiers, fichiers à votre convenance pour gérer l'intérieur de ce sous-site.

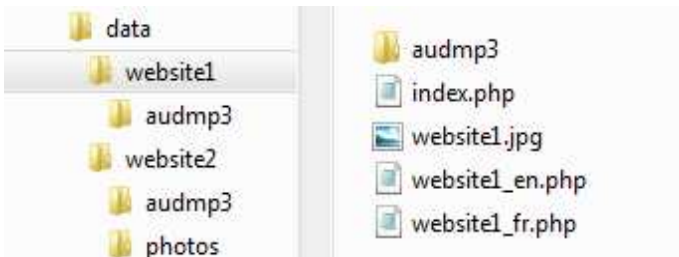


Illustration 2: Contenu d'un répertoire sous-site

Les catégories

Chaque sous-site contient des catégories. Ces dernières correspondent à une page web.

Exemple d'accès direct à une page web d'un sous-site :

« [http://\[Serveur\]/?site=\[NomSousSite\]&category=\[NomCategorie\]](http://[Serveur]/?site=[NomSousSite]&category=[NomCategorie]) »

Exemple concret :

« <http://pierre.contri.free.fr/?site=programming&category=casseBriques> »

La gestion des pages peut évidemment se faire par envoi de type « POST » au lieu de « GET ».

Présentation du Modèle Hiérarchie physique du Framework

| | |
|-------------------------|--------------------------|
| +---log | répertoire de log |
| +---sessions | gestion des sessions |
| \---sources | sources du Framework |
| | |
| index.php | |
| +---controls | Partie Contrôleur du |
| | modèle MVC |
| datasManagment.php | |
| documentsTypes.php | |
| [...] | |
| writemail.html | |
| +---images | Images globales aux |
| | différents sous-sites |
| [...] | |
| index.php | |
| ProgressBar.gif | |
| progress_bar.gif | |
| sablier.jpg | |
| [...] | |
| +---scripts | Script JavaScript pour |
| | la gestion des sites |
| includeJavaScript.php | côté client |
| index.php | |
| musicManagement.js | |
| [...] | |
| \---views | Partie Vue du modèle MVC |
| | |
| index.php | |
| masterpage.php | |
| multimediaManagment.php | |
| tabsMenuesManagment.php | |
| +---stylesheets | Feuilles de style |
| | définies par |
| global.css | l'utilisateur et le |
| index.php | développeur |
| print.css | |
| sitechoosing.css | |
| website1.css | |
| website2.css | |
| [...] | |
| \---stylesheets_old_ie | Dossier |
| | stylesheets_old_ie |
| | identique à celui |
| | précédent mais pour des |
| | versions < IE9 |

Explication principale des différents répertoires et fichiers :

/(..)*/index.php => fichier de renvoi d'erreur pour empêcher les accès manuels

./data => dossier contenant les données utilisateur (bases de données en CSV, ainsi que contenu HTML de l'utilisateur)

./data/global_nav.php => fichier permettant de générer le menu global de choix des sites, ainsi que les sitemap rattachés

./data/sitechoosing.php => fichier de choix des sous-sites

./data/siteTemplate.php.txt => fichier exemple servant de patron à la création de tout nouveau sous-site.

./log => contient les logs de demande d'accès aux sous-sites

./sessions => dossier contenant les sessions côté serveur des différents utilisateurs. Certains fournisseurs d'accès imposent ce dossier dans le site web des développeurs.

./sources => dossier contenant tout le Framework

./sources/controls => partie contrôleur : interne au Framework

./sources/scripts => partie des scripts côté client : interne au Framework

./sources/images => partie pour le développeur. Ce dossier peut contenir des images communes aux différents sites, notamment des images liées à des boutons de contrôle sur l'interface graphique

./sources/views=> contient la partie affichage du Framework

./sources/views/masterpage.php => cette page sera appelée à chaque affichage d'un site

./sources/views/stylesheet => dossier contenant chaque fichier « css » relatif à un sous-site. Il contient aussi des dossiers d'affichage de parties communes, tels que la gestion des menus et d'une télécommande multimédia.

./sources/views/stylesheet_old_ie => dossier identique au précédent, mais contenant des feuilles de style écrites pour des navigateurs Microsoft® Internet Explorer plus anciens que la version 9.

Le système de Base de Données du Framework

Le système de Base de Données du Framework est le plus simple et le plus vieux du monde : la gestion par fichier plat. En effet, collecter des données et les regrouper dans un endroit (index ou autre) n'est pas forcément lié à un système et un moteur de base de données.

Certains vrais moteurs peuvent même offrir la possibilité de stocker les données en CSV (exemple MySQL) ; c'est aussi le cas dans ce Framework.

Pour chaque site, il peut y avoir un ou plusieurs fichiers CSV permettant de définir une structure de document. Cette structure, correspondant maintenant à une classe nommée « Document », permet d'utiliser une fonction de recherche et de récupération de l'information dans la page finale du développeur en une ligne de code.

Exemple d'un fichier CSV de gestion de contenu de données :

```
//fichier texte des morceaux
//colonne 1 id
//colonne 2 titre
//colonne 3 chemin
//colonne 4 commentaires
//colonne 5 sujet

01 | Titre Chanson 1 | ./data/websitel/audmp3/autres/machanson1.mp3
| commentaire chanson | mot-cle recherche
02 | Titre Chanson 2 | ./data/websitel/audmp3/autres/machanson2.mp3
| commentaire chanson | mot-cle recherche
03 | Titre Chanson 3 | ./data/websitel/audmp3/autres/machanson3.mp3
| commentaire chanson | mot-cle recherche
04 | Titre Chanson 4 | ./data/websitel/audmp3/autres/machanson4.mp3
| commentaire chanson | mot-cle recherche
05 | Titre Chanson 5 | ./data/websitel/audmp3/autres/machanson5.mp3
| commentaire chanson | mot-cle recherche
06 | Titre Chanson 6 | ./data/websitel/audmp3/autres/machanson6.mp3
| commentaire chanson | mot-cle recherche
```

La gestion des scripts clients JavaScript

MusicManagement.js

Ce fichier contient une seule fonction permettant de gérer l'ouverture d'une fenêtre pop-up pour un morceau de musique.

```
Function listen(fileSoundName,soundName,soundComment) ;
```

paramètres d'entrée :

- fileSoundName : chemin web du son ou de la musique à écouter
- soundName : titre du morceau
- soundComment : commentaires liés à cette musique

résultat de sortie :

- fenêtre pop-up contenant le nom du morceau, son commentaire, ainsi qu'un lecteur média exécutant la musique inscrite dans le chemin web

PictureManagement.js

Ce fichier est bien plus complexe que celui lié à la gestion audio car celle de l'image dans les différents navigateurs web nécessite plus de code.

Le fichier commence par la définition de l'index du type tableau si ce dernier n'est pas implémenté dans les différentes versions des navigateurs.

```
function getSizeScreen()
```

Permet de récupérer la taille de l'écran.

Retourne une structure (x ; y) contenant la largeur et la hauteur de l'écran.

```
function getSizeScreenDoc(docVar)
```

Permet de récupérer la taille du document passé en paramètres. Cela peut être un cadre à l'intérieur du navigateur web.

En entrée : objet Javascript à analyser.

Retourne une structure (x ; y) contenant la largeur et la hauteur de l'objet demandé.

```
function getSizeInnerScreen()
```

Permet de récupérer la taille de l'écran.

Retourne une structure (x ; y) contenant la largeur et la hauteur de l'écran.

```
function getScrollXY()
```

Permet de récupérer la position des ascenseurs de défilement de la page web.

Retourne une structure (x ; y) contenant le scroll top et le scroll left de la page web.

```
function switchDiaporama()
```

Permet de changer l'affichage des photos d'une page. Passage de diaporama en tableau et réciproquement.

```
function print_picture(nomPhoto)
```

Affiche une photo sur une fenêtre de type « pop-up ».

En entrée : chemin web de la photo à afficher.

myMiniFramework for meta web site

Mini Framework pour site web basé sur PHP

```
function load_picture(nameImgContenuPhoto, nameInputListePhotos,
sens)
```

Affiche une photo dans un cadre défini par rapport à une liste pré-définie avec les objets complexes de classe « Document » et « DocumentsList ».

Paramètres d'entrée :

- nameImgContenuPhoto : nom de la photo à afficher
- nameInputListePhotos : nom des balises de type « input hidden » définissant la liste des différentes photos à afficher contenant le nom de la photo, ainsi que le chemin web
- sens : permet de demander à la fonction le sens de récupération des photos dans la liste précédente ('random', 'next', 'previous', 'first', 'last')

Retourne le résultat de bon fonctionnement de la fonction.

```
function show_picture_fullScreen(pictureName, pictureAlt)
```

Permet d'afficher en plein écran une photo et son intitulé.

En entrée : la source de l'image, la légende de l'image.

```
function defilStartStop(nameButtonDefil)
```

Démarre ou arrête (suivant l'état précédent) le diaporama des images contenu dans les « input hidden » de la page web.

paramètre d'entrée : identifiant de la balise HTML du bouton de démarrage/arrêt du diaporama. Cela permet de changer de signe de fonctionnement sur le bouton.

```
function updatePictureNumber(listByName, pictureName)
```

Met à jour le numéro de la photo dans la barre de défilement.

En entrée : le nom de la balise « input hidden » contenant la liste des photos, le nom de la photo actuellement sélectionnée

```
function getObjectIndexInList(listByName, searchName)
```

Retourne l'index auquel se trouve la place du nom cherché dans une liste de « input hidden »

En entrée : le nom de la balise « input hidden » contenant la liste des photos, le nom de l'objet à chercher

En sortie : index de l'objet dans le tableau


```
function getNextPicture(pictureList, pictureName)
```

Récupère le nom de la photo suivante fourni dans la liste par rapport à la photo actuelle.

En entrée : le nom de la balise « input hidden » contenant la liste des photos, le nom de la photo actuelle servant de base de recherche.

Retourne le nom de la photo suivante.

```
function getPreviousPicture(pictureList, pictureName)
```

Récupère le nom de la photo précédente fourni dans la liste par rapport à la photo actuelle.

En entrée : le nom de la balise « input hidden » contenant la liste des photos, le nom de la photo actuelle servant de base de recherche.

Retourne le nom de la photo précédente.

```
function getRandomPicture(pictureList)
```

Récupère le nom d'une photo contenu dans la liste fournie.

En entrée : le nom de la balise « input hidden » contenant la liste des photos.

Retourne le nom de la photo choisie aléatoirement.

SiteManagement.js

`var parseXml`

Création d'un objet permettant de parser du XML suivant différents navigateurs web

`function refreshPage()`

Permet de recharger la page web actuelle par un envoi « POST ».

`function goCategory(categoryName)`

Permet de passer à une nouvelle page web (du même sous-site web).

En entrée : nom d'une page web du sous-site sélectionné.

`function goSite(sitepath)`

Permet de passer à un autre sous-site ou d'entrer dans un sous-site si l'on est à la racine.

En entrée : nom du sous-site.

`function getCategoryWebS(categoryName)`

Permet de récupérer une page web (du même sous-site web) par appel client.

En entrée : nom d'une page web du sous-site sélectionné.

Retourne le contenu web de la page demandée.

`function show_div(divParent), function hide_div(divParent)`

Permet d'afficher ou de masquer un div.

En entrée : l'identifiant du div à gérer (<div id='monDiv'></div>).

`function postMail()`

Permet d'envoyer en POST le mail soumis dans une page web.

`function getAbsolutePosition(element)`

Récupère la position absolue dans la feuille web d'un élément.

En entrée : élément à chercher dans la feuille.

`Function moveIcôneByMouse()`

Permet de déplacer un élément nommé « icône » avec le pointeur de souris.

WaitingManagement.js

```
function patiente()
```

Affiche un sablier et demande à l'utilisateur de patienter le temps de chargement de la page.

```
function progressOpacity(elementDivName, tpsTimeout, opacityStart,  
opacityEnd)
```

Affiche progressivement un fond noir sur la page web actuelle afin de mettre en avant le sablier d'attente.

```
function testAffichageDivPatienter()
```

Permet au développeur de faire une démonstration de la demande d'attente.

```
function cachePatienter()
```

Cache le sablier d'attente et rend la main à l'utilisateur.

```
function affichePatienter()
```

Affiche le sablier et prend la main sur l'affichage.

```
function createPatienter()
```

Permet de créer automatiquement la partie « patienter » dans n'importe quelle page web.

Créer un site (ou sous-site) web

Contenu de ce chapitre

| | |
|--|----|
| Créer un site (ou sous-site) web..... | 35 |
| Le fichier patron de création d'un sous-site..... | 36 |
| Étapes de création d'un site web avec le Framework..... | 37 |
| Créer un menu pour un sous-site..... | 38 |
| La balise « entry »..... | 39 |
| La balise « menu »..... | 39 |
| Utiliser la base de données pour gérer les objets multimédias..... | 40 |

Le fichier patron de création d'un sous-site

Présentation du fichier « patron » permettant de créer un sous-site (copier-coller).

| Code « siteTemplate.php.txt » | Remarques |
|--|---|
| <pre><?php // ----- // Differents variables for the web site \$_SESSION["dirPhotos"] = dirname(\$_SESSION['sheetname']) . "/photos"; \$_SESSION["dirDocuments"] = "./data/website/presentations"; \$_SESSION["dirTools"] = "./utility"; \$_SESSION["ficPhotos"] = \$_SESSION["dirPhotos"] . "/photos_en.csv"; // ----- // ----- // Menu if(!isset(\$_SESSION["xmlMenu"])) \$_SESSION["xmlMenu"] = <<<EndMenu <menu name="main"> <entry name="" visible="true" default="true"> <titlePage></titlePage> </entry> <entry name="" visible="true"> <titlePage></titlePage> <menu name=""> <entry name="" visible="true"> <titlePage></titlePage> <menu></menu> </entry> </menu> </entry> </menu> EndMenu; // -----</pre> | <p>Partie définition des variables session. Emplacement des chemins de base de données de documents, ...</p> <p>Création du menu et mise en mémoire de ce dernier pendant toute la navigation dans le sous-site</p> |

```
// Web pages content
```

```
function categoryName() {  
    $webPage = new WebPage();  
    $webPage->docTitle      = "";  
    $webPage->metaWords     = "";  
    $webPage->contentPage   = "";  
    return $webPage;  
}
```

```
// -----  
?>
```

Pour chaque
entrée du menu,
une fonction
retournant une
page web
« WebPage ».

Étapes de création d'un site web avec le Framework

- 1) Créez un dossier avec le nom du site dans le dossier data. (le nom du site sera « foo » pour l'explication)
- 2) Dans le dossier ./data/foo, créez un fichier foo.php si vous ne le voulez que dans une seule langue, sinon, créez autant de fichier foo_[...].php que de langues que vous souhaitez. Par exemple « foo_en.php », « foo_fr.ph »
- 3) Copiez le contenu du fichier « siteTemplate.php.txt » dans votre fichier « foo.php »
- 4) Créez le fichier de style nommé « foo.css » dans le dossier ./sources/views/stylesheet. Définissez le type de menu (vertical, horizontal) les couleurs de votre site, la présentation, ...
- 5) Modifiez par votre contenu le fichier « foo.php »
- 6) Testez, puis n'utilisez que le fichier foo.php pour ajouter des catégories.

Créer un menu pour un sous-site

Dans chaque sous-site (fichier « foo.php ») se trouve une rubrique « xmlMenu ». Cette rubrique est constituée d'une chaîne de caractères en format XML.

```
// -----  
// Menu  
if(!isset($_SESSION["xmlMenu"]))  
$_SESSION["xmlMenu"] = <<<EndMenu  
<menu name="main">  
  <entry name="" visible="true" default="true">  
    <titlePage></titlePage>  
  </entry>  
  
  <entry name="" visible="true">  
    <titlePage></titlePage>  
    <menu name="">  
      <entry name="" visible="true">  
        <titlePage></titlePage>  
        <menu></menu>  
      </entry>  
    </menu>  
  </entry>  
</menu>  
  
EndMenu;
```

Le code correspondant au menu du sous-site comprend deux types de balises :

- la balise « entry »
- la balise « menu »

La balise « entry »

La balise « entry » peut recevoir trois attributs :

- name : le nom de l'entrée correspondant aussi à la catégorie sélectionnée dans la page web
- visible : la possibilité d'afficher ou nom l'entrée à l'écran. Sachant qu'une entrée doit exister pour être appelée par le contrôleur principal du site (gestion des catégories)
- default : la possibilité de déclarer cette entrée (ou catégorie) comme page par défaut si l'utilisateur n'en a pas sélectionné explicitement

La balise « entry » peut recevoir deux types de balises filles :

- la balise « titlePage » contenant le titre de la page à afficher dans le menu utilisateur
- la balise « menu » pouvant contenir un sous-menu dans cette catégorie

La balise « menu »

La balise « menu » est un conteneur de balise « entry ».

La balise « menu » ne peut recevoir qu'un seul type de balise : la balise « entry ». Il est, bien sûr, possible de positionner plusieurs balises « entry » à l'intérieur d'une balise « menu », puisque c'est un conteneur.

La balise « menu » accepte un argument. Ce dernier est « name » et correspond au nom de la catégorie parente de ce menu. Quand le menu est le « root », son nom est « main », mais à ce niveau, ce dernier n'a aucune importance.

Une explication graphique plus approfondie se trouve dans le « Diagramme de classe du menu ».

Utiliser la base de données pour gérer les objets multimédias

Présenté précédemment, le système de base de données du Framework est archaïque, mais fonctionne sans bug.

Vous pouvez créer un ou plusieurs fichiers CSV.

Dans chacun de ceux-ci, vous devez respecter le format de 4 ou 5 colonnes, suivant le type d'affichage que vous souhaitez.

| Description des colonnes | Remarques |
|--|---|
| //colonne 1 id //colonne 2 titre //colonne 3 chemin //colonne 4 commentaires //colonne 5 sujet | Seule une colonne fait la différence entre les deux types de format. Pour certains types de fichiers, il n'est pas nécessaire de posséder un commentaire (par exemple un type photo), tandis que d'autres types (audio) peuvent contenir un commentaire. Il est donc possible de choisir entre les deux. Par doute, nous pouvons tout placer avec des commentaires, et laisser le champ vide s'il n'y a rien. |
| // colonne 1 id // colonne 2 chemin // colonne 3 commentaires // colonne 4 sujet | |

Votre fichier CSV peut contenir des commentaires.
Il faut alors commencer la ligne par « // ».

La dernière colonne correspond à l'index de recherche de toute demande dans cette base de données par la fonction « getListElems([nomFichier], [sujet]) ».

Le séparateur de colonne de ces fichiers CSV est le « pipe » ou « | ».

Après avoir créé votre fichier CSV (cf exemple au chapitre « Le Système de Base de Données du Framework »), vous devez renseigner le chemin du fichier CSV dans votre fichier

Par exemple, dans votre fichier « ./data/foo/foo.php » :

```
<?php
// Mon site musical en php
// Pierre Contri
// Cree le 02/11/2004
// modifie le 12/02/2010
// Editeur : Notepad
// Content the data of this site

[...]

// base de données photos
$_SESSION["ficPhotos"]      = "./data/foo/photos_en.csv";
// base de données audio
$_SESSION["ficMorceaux"]    = "./data/foo/morceaux_en.csv";

[...]

// Utilisation de cette base
maNouvelleCategorie() {
    $listPhotos = getListElems($_SESSION["ficPhotos"], "*") ;
    [...]
}
```



Illustration 3: Exemple d'utilisation multiple de la fonction "getListElems"

Les diagrammes UML simplifiés du Framework

Contenu de ce chapitre

| | |
|---|----|
| Les diagrammes UML simplifiés du Framework..... | 43 |
| Le diagramme de classe global..... | 44 |
| Le diagramme de classe global (suite)..... | 45 |
| Les classes Documents..... | 46 |
| Le diagramme de classe de la fabrique de documents..... | 48 |
| Le diagramme de classe des pages Web..... | 49 |
| Le diagramme de classe du menu..... | 50 |

Le diagramme de classe global

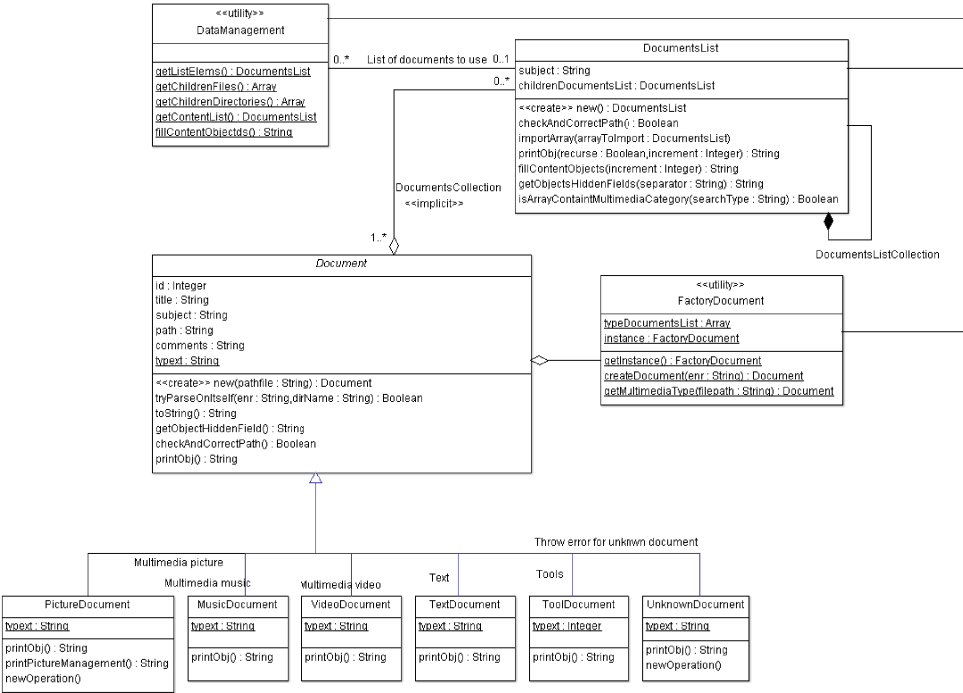


Illustration 4: Diagramme de classe global du Framework

Le diagramme de classe global (suite)

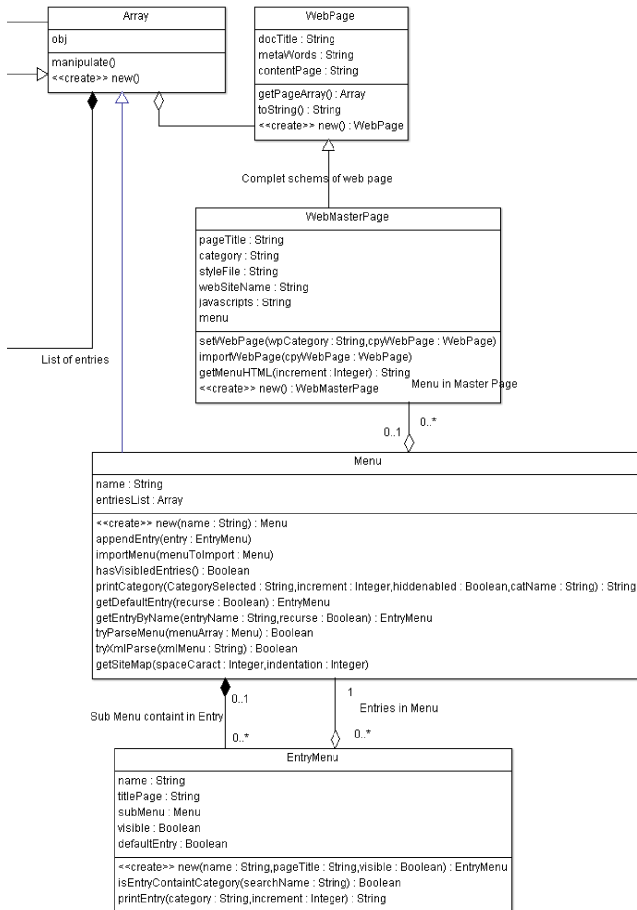


Illustration 5: Suite du diagramme de classe global

Les classes Documents

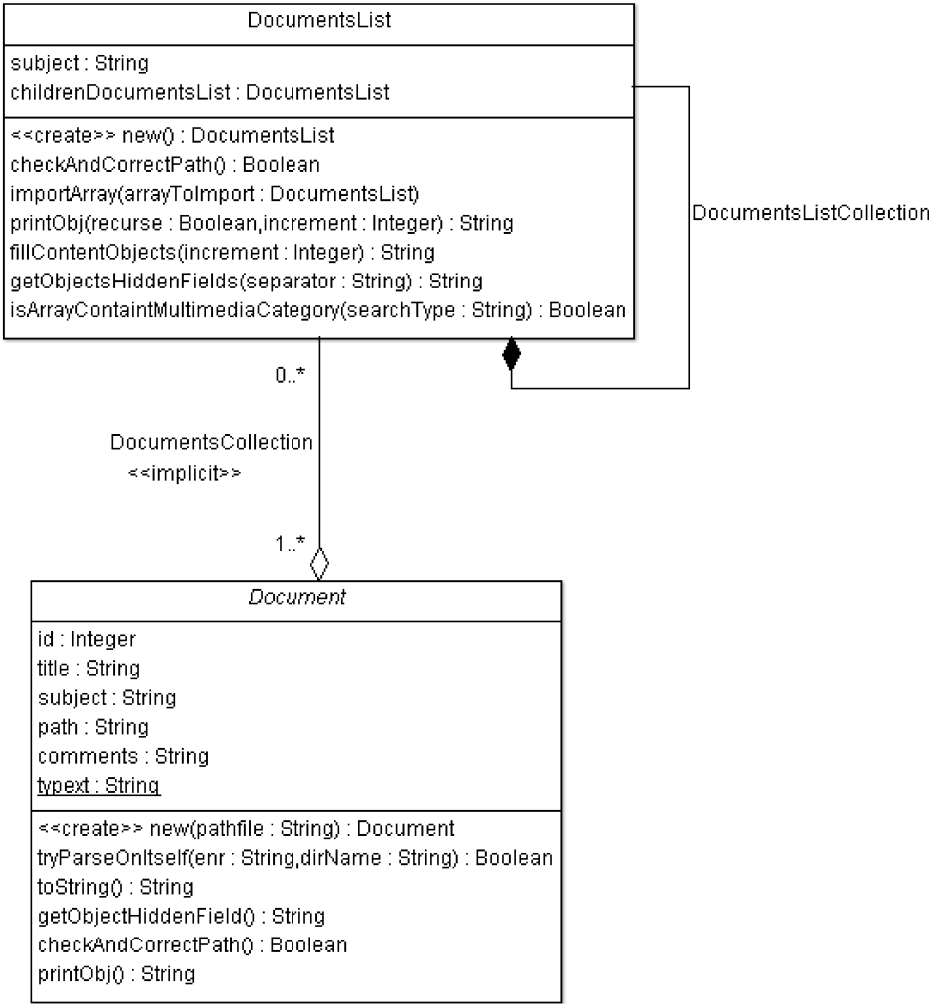


Illustration 6: Diagramme de classe des Documents (vue générique)

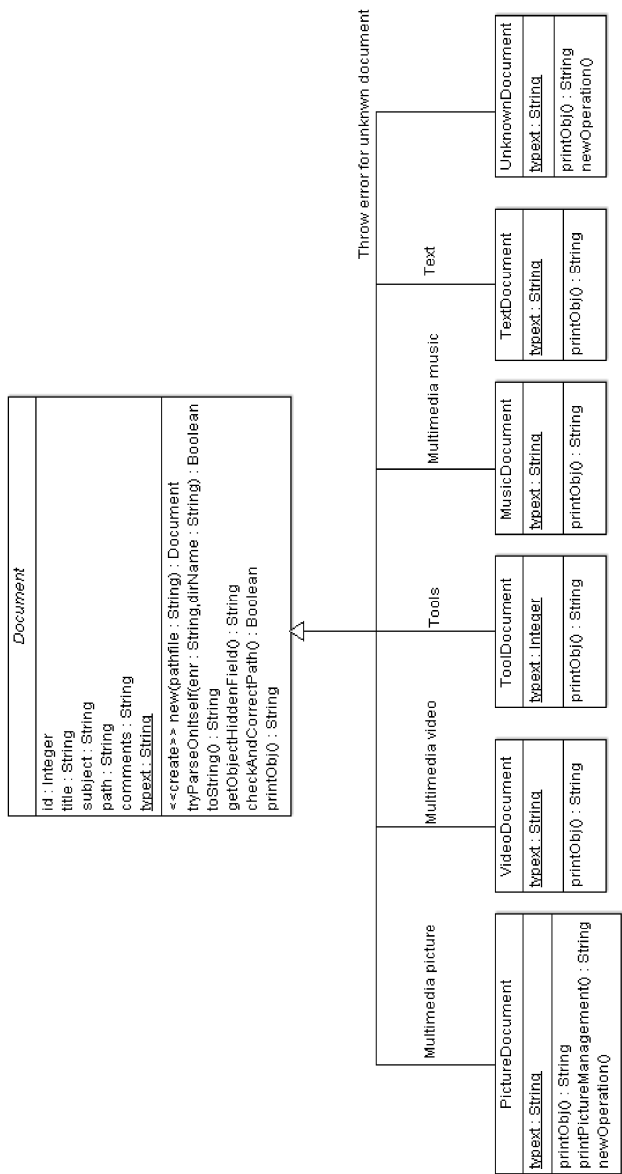


Illustration 7: Extension des diagrammes de classe

Le diagramme de classe de la fabrique de documents

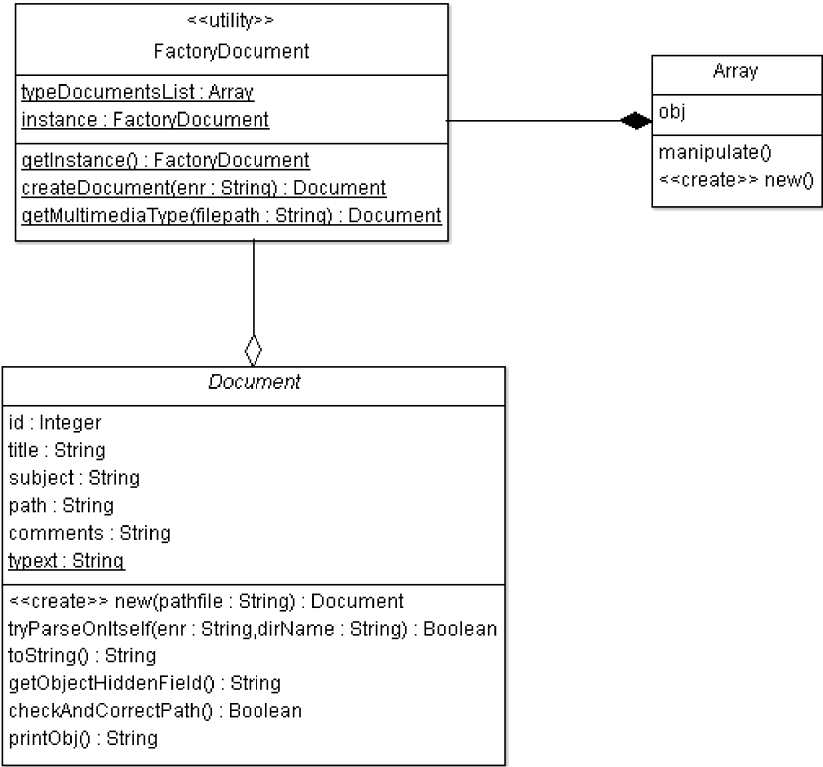


Illustration 8: Diagramme de classe de la fabrique de documents

Le diagramme de classe des pages Web

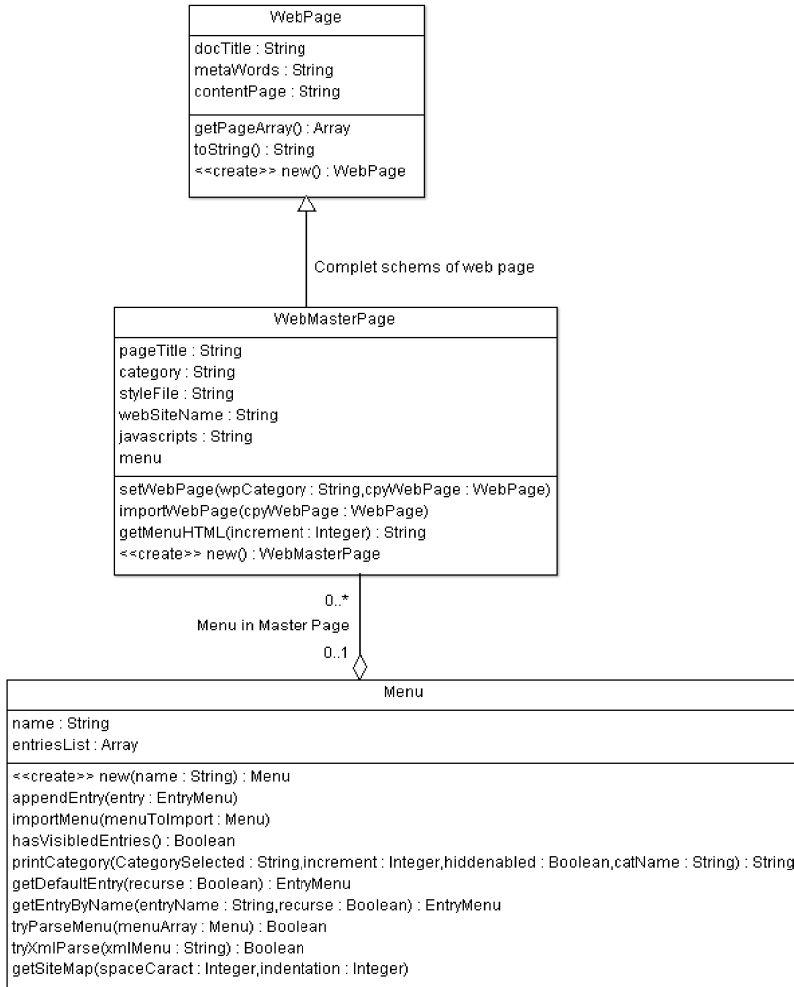


Illustration 9: Diagramme de classe des pages web

Le diagramme de classe du menu

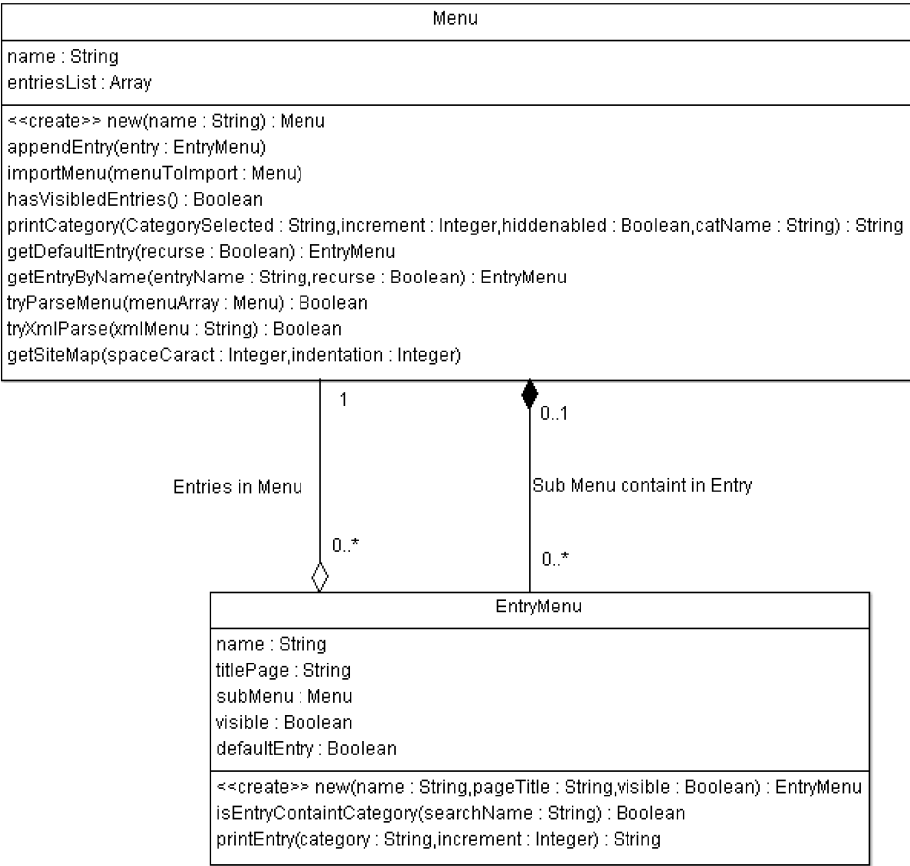


Illustration 10: Diagramme de classe de la partie Menu

Les fonctions principales du Framework

Contenu de ce chapitre

| | |
|--|----|
| Les fonctions principales du Framework..... | 51 |
| Fonctions basiques du Framework..... | 52 |
| Fonction liée à la base de données du Framework..... | 52 |
| Fonction liée à la structure du serveur web physique..... | 53 |
| Fonctions liées aux répertoires et fichiers de l'ordinateur..... | 54 |
| Fonctions avancées du Framework..... | 55 |
| Description..... | 55 |
| Les deux objets à utiliser : « DocumentsList » et « Document »..... | 56 |
| DocumentsList..... | 56 |
| Document..... | 58 |
| La fabrique de documents (« FactoryDocument »)..... | 60 |
| Utilisation des fonctions du Framework..... | 62 |
| Utilisation de l'affichage d'un tableau de photos dans une page web par l'exemple..... | 62 |
| Utilisation de l'affichage d'un tableau de morceaux audio dans une page web par l'exemple..... | 64 |
| Utilisation mélangée du Framework et d'une partie HTML..... | 66 |

Fonctions basiques du Framework

Fonction liée à la base de données du Framework

```
/**
 * Function : getListElems
 *   Browse a csv file and return an array
 *   with the selected subject
 * Inputs : - database file name (string)
 *           - subject (optional string) filter
 * Output : - infos array of Documents
 * Author : Pierre Contri
 */
function getListElems($ficNameListElem, $subject = "")
```

La fonction « `getListElems` » permet de récupérer les éléments filtrés (par le paramètre « `subject` ») de la base de données (fournie en lien par le paramètre « `ficNameListElem` »). Le retour de cette fonction est un objet de type `DocumentsList` qui hérite d'un tableau et permet de fournir une liste de type `Documents`. Ces documents peuvent, bien sûr, être de différentes natures (texte, musique, images, ...).

L'objet retourné a les capacités de fournir du code HTML pouvant directement s'inscrire dans la page (cf exemples d'affichage d'un tableau de photos).

Exemple simple :

```
<?php
include "../sources/controls/dataManagement.php" ;

$listElems = getListElems("[fichierBDD].csv", "photos_concert") ;
print($listElems[0]->printObj()); // affichage d'une photo
print($listElems->printObj()); // affichage du tableau de photos

?>
```

Fonction liée à la structure du serveur web physique

```
/**
 * Function :
 *
 * Inputs : - directory path (string)
 *          - type of document (subject in the CSV DataBase)
 * Output : - infos array of Documents (DocumentsList)
 * Author : Pierre Contri
 */
function getContentList($repertoire, $type = "*", $recursive =
true)
```

La fonction « `getContentList` » est un peu différente de la fonction précédente, mais garde la même philosophie. Il s'agit, non pas à partir d'une base de données, mais d'un répertoire physique, de récupérer une liste de documents (fournie également avec un objet « `DocumentsList` »), en précisant le répertoire (paramètre « `repertoire` »), le type (l'extension du type de fichier par le paramètre « `type` »), ainsi que la possibilité de descendre en récursivité dans les répertoires enfants.

Exemple simple :

```
<?php
include "../sources/controls/dataManagement.php" ;

$listElems = getListElems("[monRepertoirePhoto]", "jpg") ;
print($listElems[0]->printObj()); // affichage d'une photo
print($listElems->printObj()); // affichage du tableau de photos

?>
```

Fonctions liées aux répertoires et fichiers de l'ordinateur

```
/**
 * Function : getChildrenFiles
 *   Open the directory in parameters
 *   browse it, read it and get content
 * Inputs : - directory path (string)
 * Output  : - array of directory content (only files)
 * Author  : Pierre Contri
 */
function getChildrenFiles($directoryPath)
```

```
/**
 * Function : getChildrenDirectories
 *   Open the directory in parameters
 *   browse it, read it and get content
 * Inputs : - directory path (string)
 * Output  : - array of directory content (only directories)
 * Author  : Pierre Contri
 */
function getChildDirectories($directoryPath)
```

Ces deux fonctions ont été redéfinies par rapport à PHP pour des raisons pratiques et fonctionnelles liées au site. L'un récupère la liste des fichiers contenu dans un répertoire, l'autre récupère la liste des dossiers d'un répertoire. Le tout est rendu dans un « array ».

Fonctions avancées du Framework

Description

Le Framework propose une encapsulation de code dans les classes « Document » et « DocumentsList » qui permettent, par exemple :

- si le document est de type image ;
 - de proposer directement un diaporama
 - de fournir du code permettant de cliquer sur l'image pour l'afficher en plein écran
- si le document est de type audio ;
 - de proposer un lecteur audio permettant de le lire
 - de proposer le téléchargement du titre
- [...]

Le principe est bien d'encapsuler tout document et de demander aux différentes classes d'afficher un type ou un autre sans avoir à se soucier du contenu du document.

Cette partie avancée du Framework est incluse dans les classes dérivant la classe « Document ». Une partie du code est donc réalisée aussi côté client web. Cela veut dire que l'on inclut aussi des pages JavaScript de manière automatique chez le client.

Nota Bene : bien que j'ai beaucoup de respect pour « jQuery » (né en 2006), ce dernier est apparu bien après le début de ce travail. En milieu 2005, toute la partie JavaScript était déjà réalisée, c'est aussi la raison pour laquelle je n'ai pas utilisé jQuery dans mon développement client.

Les deux objets à utiliser : « DocumentsList » et « Document »

DocumentsList

L'objet « DocumentsList » est retourné par les fonctions dites « interface » du Framework. Cet objet complexe a la possibilité de renvoyer le code HTML de chaque Document qu'il contient. Le jeu avec les tableaux est très agréable en PHP et est des plus facile à réaliser. « DocumentsList » hérite donc de « ArrayObject » de PHP. Il n'y a que très peu de travail à réaliser sur cette classe, tant la qualité et la pluralité de possibilité de cette classe est importante. Nous ne pouvons que nous incliner sur la qualité de PHP (Existe aussi avec DotNet et Python).

J'ai enrichi le modèle de « ArrayObject » de cinq méthodes de classe (plus le constructeur) et deux variables.

Les variables :

subject : le sujet du « DocumentsList »

childrenDocumentsList : un « DocumentsList » fils, ce qui permet d'imbriquer cette classe sur elle-même.

Les méthodes :

```
__construct($array = array())
```

Le constructeur est redéfini, car il peut recevoir un tableau en paramètres. Si tel est le cas, le contenu de ce tableau sera inséré dans ce nouveau « DocumentsList »

```
importArray($tmpArray)
```

Copie le contenu du tableau passé en paramètres dans l'objet actuel.

```
printObj($recurse = true, $increment = 4)
```

Retourne du code HTML permettant d'afficher le tableau de document que l'objet contient. Pour cela, il fera appel aux différents « Document » en leur demandant de renvoyer leur propre contenu HTML pour l'affichage. Le contenu sera fourni sous forme de liste HTML

```
fillContentObjects($increment = 4)
```

Retourne du code HTML permettant d'afficher le tableau de document que l'objet contient. Pour cela, il fera appel aux différents « Document » en leur demandant de renvoyer leur propre contenu HTML pour l'affichage. Le contenu sera fourni sous forme d'un « DIV » lui-même contenant plusieurs « DIV » de « Document ».

Le paramètre « increment » permet simplement de rendre le code HTML indenté pour qu'il soit lisible au niveau du client. Attention tout de même à passer « 0 » en tout HTML5, car les espaces sont pour l'instant déconseillés à certains endroits de la page web.

```
getObjectsHiddenFields($separator = "")
```

Cette méthode retourne du code HTML contenant la liste des documents internes comme référence pour le Framework en balise `<input type="hidden" />`

Cela est utilisé pour faire le lien avec la partie JavaScript afin d'afficher une ou plusieurs images, ainsi que gérer une liste de lecture audio, ...

```
isArrayContaintMultimediaCategory($searchType)
```

Cette méthode permet de détecter si la collection de documents qu'elle stocke comporte au moins un document du type demandé en paramètres.

Par exemple, il est pratique de demander à l'objet si ce dernier contient des images. Dans ce cas, l'objet pourra préparer la télécommande de consultation d'images, ainsi que le diaporama en arrière plan.

Document

La classe « Document » est le pivot de ce Framework. C'est sur cette classe abstraite que tout repose. Elle offre cinq variables d'instance, une variable de classe, quatre méthodes et deux redéfinitions de méthodes de l'objet de base.

L'objet « Document » représente les données de chaque ligne des différentes bases de données CSV.

Les variables d'instance :

id : index de l'objet

title : titre du document

subject : sujet, correspond à la catégorie de recherche dans la base CSV

path : chemin web du document lui-même

comment : commentaires (facultatif)

La variable de classe :

typext : chaîne de caractères contenant la liste des extensions possibles d'un document. Bien entendu, le « Document » étant abstrait, sa chaîne contenant les extensions des chemins de document est vide. Elle doit être redéfinie par les classes dérivées.

Les méthodes redéfinies de l'objet de base :

`__construct($pathfile = "")`

Le constructeur peut recevoir le chemin web de l'objet en chaîne de caractères.

`__toString()`

Permet de redéfinir le rendu de la classe « Document ». Par exemple, lorsque l'on crée un objet Document, on peut directement demander à la fonction « print » de PHP de l'afficher.

Les méthodes de classe :

```
tryParseOnItself($enr, $dirName = ".")
```

Cette méthode permet, à partir d'une ligne d'un enregistrement (« enr ») de la base de données CSV, de récupérer les informations du document et de les enregistrer sur ses variables de classe. Le seconde paramètre (« dirName ») permet de concaténer un chemin relatif contenu dans la base de données, avec le chemin relatif du sous-site si besoin afin de le transformer en absolu. Sinon, le « path » restera relatif.

```
getObjectHiddenField()
```

Retourne le code HTML d'une balise de type « input hidden » contenant les informations de l'objet pour le Framework.

```
checkAndCorrectPath()
```

Cette méthode permet de vérifier que le chemin relatif au document est correct. Si le document a des différences au niveau de la « casse » (Majuscules / Minuscules), cette dernière modifiera la chaîne du chemin. Elle retournera la confirmation de vérification de l'existence du chemin.

```
printObj()
```

Retourne le codage HTML de l'objet « Document ». Cette méthode doit être redéfinie par tous les documents dérivés.

La fabrique de documents (« FactoryDocument »)

Cette classe permet au Framework lui même de s'appuyer sur un créateur de document. En effet, lorsque l'on scrute la base de données, on connaît le type de document par l'extension de son chemin. Il est alors hors de question d'aller, dans chaque récupération de document, demander à chaque classe « est-ce que ce document est de votre type ? ».

Pour cela, j'ai créé une fabrique de documents qui va questionner tous les documents présents.

A l'heure actuelle, le Framework est compatible PHP 5.1. Il ne m'a pas été possible d'écrire du code pour lui demander la liste de toutes les classes dérivant « Document », ainsi que pour lui dire de demander à chacune d'entre-elle si le document est de leur type.

Je sais le faire avec les avancées de PHP, notamment PHP 5.3. Par souci de compatibilité avec certains hébergeurs, ce code est désactivé, et une partie moins classe est rajoutée pour compenser.

Cette fabrique est un singleton, mais son implémentation va encore être améliorée car sa qualité de codage, quoique efficace, ne me plaît pas. Cela mérite mieux.

Deux méthodes de classe :

```
getMultimediaType($filepath)
```

En fonction du chemin du fichier passé en paramètres, c'est cette méthode qui va déterminer quel est le type de document à utiliser pour ce futur objet.

```
createDocument($enr)
```

S'appuyant sur la méthode précédente, cette méthode va appeler la classe définie pour créer le document. Cela permettra aussi de passer en paramètres le chemin du document. Son paramètre d'entrée peut être soit un tableau (une ligne d'un enregistrement CSV), soit un chemin en chaîne de caractères.

Utilisation des fonctions du Framework

Utilisation de l'affichage d'un tableau de photos dans une page web par l'exemple

```
function photoBrest() {  
    $listElems = getListElems($_SESSION["ficPhotos"], "Brest");  
    $webPage = new WebPage();  
    $webPage->docTitle      = "Brest et environs";  
    $webPage->metaWords     = "aperçu de Brest, Port de Brest,  
Ch&acirc;teau de la Marine";  
    $webPage->contentPage = $listElems->fillContentObjects();  
    return $webPage;  
}
```

Explication de l'exemple

- 1) Définition de la fonction représentant une catégorie du site (« function photoBrest »)
- 2) Récupération d'une liste de documents de type PictureDocuments (« \$listElems = getListElems(\$_SESSION["ficPhotos"], "Brest"); »)
- 3) Création d'une page web (« \$webPage = new WebPage(); »)
- 4) Renseignement du titre de la page (« \$webPage->docTitle = "Brest et environs"; »)
- 5) Renseignement des mots clés de cette page web (« \$webPage->metaWords = "aperçu de Brest, Port de Brest, Château de la Marine"; »)
- 6) Création du contenu de la page (« \$webPage->contentPage = \$listElems->fillContentObjects(); »). Ce contenu est très simple, puisque l'on demande à l'objet (qui est une liste de documents) de s'afficher.
- 7) Renvoyer la page pour qu'elle puisse être insérée dans la MasterPage

Exemple :

« <http://pierre.contri.free.fr/?site=instrumental&category=photoBrest> »

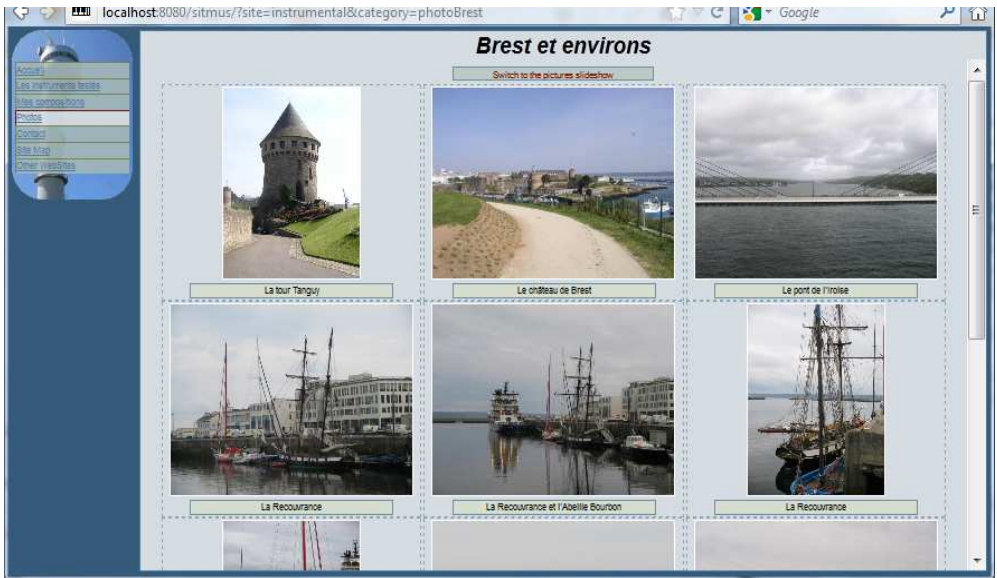


Illustration 11: Création d'une page web affichant un tableau d'images

Utilisation de l'affichage d'un tableau de morceaux audio dans une page web par l'exemple

```
function morceauxEsquisse() {  
    $listElems = getListElems($_SESSION["ficMorceaux"], "Pierre  
Contri esquisse");  
    $webPage = new WebPage();  
    $webPage->docTitle = "Petits morceaux en esquisse";  
    $webPage->metaWords = "pierre contri, sacem, morceaux de  
demonstration, musique instrumentale, compositions  
personnelles,morceaux de d&eacute;monstration";  
    $webPage->contentPage = $listElems->fillContentObjects();  
    return $webPage;  
}
```

Explication de l'exemple

- 1) Définition de la fonction représentant une catégorie du site (« fonction morceauxEsquisse»)
- 2) Récupération d'une liste de documents de type MusicDocuments (« \$listElems = getListElems(\$_SESSION["ficMorceaux"], "Pierre Contri esquisse"); »)
- 3) Création d'une page web (« \$webPage = new WebPage(); »)
- 4) Renseignement du titre de la page (« \$webPage->docTitle = "Petits morceaux en esquisse; »)
- 5) Renseignement des mots clés de cette page web (« \$webPage->metaWords = "pierre contri, morceaux de demonstration, musique instrumentale"; »)
- 6) Création du contenu de la page (« \$webPage->contentPage = \$listElems->fillContentObjects(); »). Ce contenu est très simple, puisque l'on demande à l'objet (qui est une liste de documents) de s'afficher.
- 7) Renvoyer la page pour qu'elle puisse être insérée dans la MasterPage

Exemple :

« <http://pierre.contri.free.fr/?site=instrumental&category=morceauxEsquisse> »



Illustration 12: Création d'une page web affichant un tableau de morceaux musicaux

Utilisation mélangée du Framework et d'une partie HTML

```
function CarCostSimulator() {
    $webPage = new WebPage();
    $webPage->docTitle      = "Car Cost Simulator";
    $webPage->metaWords     = "car cost simulator";
    $listElems = getListElems($_SESSION["ficPhotos"], "calculcar");
    $webPage->contentPage = <<<ENDComments
<p>Due to a need changing car, I decide to create a car cost
simulator. Many people say we need a diesel motor if we make more
kilometers than 15000 per year. We my software, I try to break this
idea.</p>
<p>On this software, we can enter every energies that you want, all
car (or trucks) and try to simulate an driving type influe on the
consumption.</p>
<p>Requierelement :</p>
<ul>
    <li>python 2.6</li>
    <li>gtk</li>
    <li>lib glade</li>
    <li>matplotlib</li>
    <li>scipy</li>
    <li>numpy</li>
    <li>pylab</li>
</ul>
<p><a href="./data/programming/CarCostSimulator.zip">Download
it</a></p>
<br />

{$listElems->fillContentObjects()}

ENDComments;

    return $webPage;
}
```

Exemple :

« <http://pierre.contri.free.fr/?site=programming&category=CarCostSimulator> »

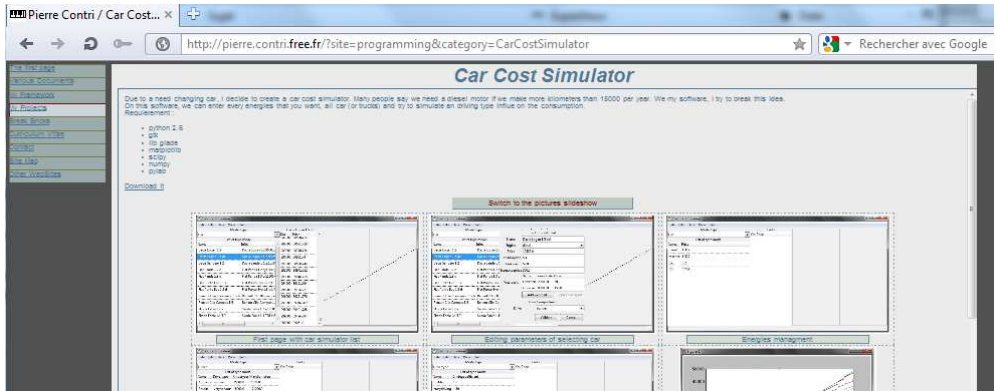


Illustration 13: Exemple en image d'une page web mélangeant contenu texte et images en base de données

Dans cet exemple, plusieurs images sont affichées. Un bouton permettant de passer d'un tableau de photos à un diaporama d'une seule photo apparaît sans ajouter de code de la part du développeur final.

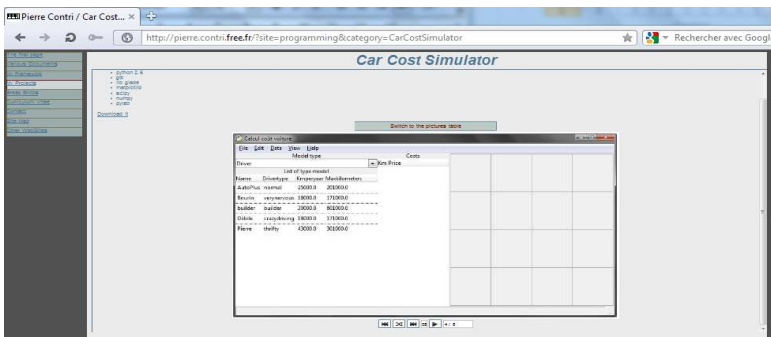


Illustration 14: Exemple de passage en mode diaporama de la page précédente

Exemple de « Hello World » avec ce Framework

Contenu de ce chapitre

| | |
|--|----|
| Exemple de « Hello World » avec ce Framework..... | 69 |
| Création du fichier de données (« helloworld.php »)..... | 70 |
| Création du fichier de style (« helloworld.css »)..... | 72 |

Création du fichier de données (« helloworld.php »)

- 1) Créez un dossier « helloworld » dans le dossier « data »
- 2) Créez ensuite le fichier « helloworld.php » dans le dossier « ./data/helloworld »
- 3) Insérez le code suivant

Code

```
<?php
// Mon site musical en php
// Pierre Contri
// Cree le 02/11/2004
// modifie le 12/02/2010
// Editeur : Notepad
// Content the data of this site

// Files database names in environment variables
// Aucune base necessaire pour l'exemple Hello World

$_SESSION["ficPhotos"]      = "../data/helloworld/photos_en.csv";
$_SESSION["ficMorceaux"]    = "../data/helloworld/morceaux_en.csv";

// -----
// menu

if(!isset($_SESSION["xmlMenu"]))
$_SESSION["xmlMenu"] = <<<EndMenu
<menu name="main">
  <entry name="helloworld" visible="true" default="true">
    <titlePage>Hello World</titlePage>
  </entry>
</menu>

EndMenu;
// -----
```



```
function helloworld() {  
    $webPage = new WebPage();  
    $webPage->docTitle      = "Hello World";  
    $webPage->metaWords     = "test myMiniFramework par Hello World";  
    $webPage->contentPage = <<<EndTxt  
<h1>Hello World !</h1>  
  
EndTxt;  
    return $webPage;  
}  
// End of content  
// -----  
?>
```

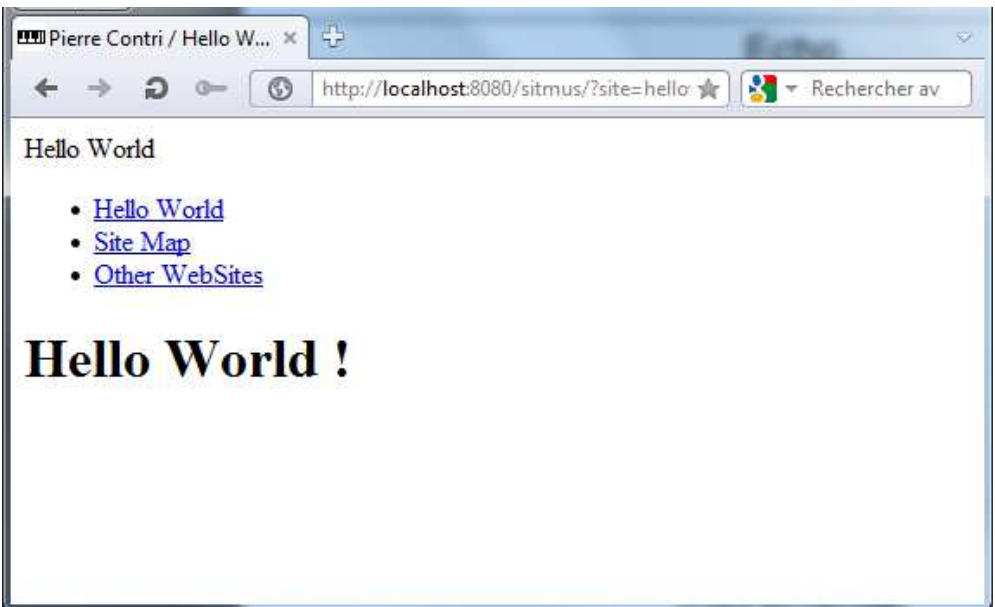


Illustration 15: Résultat graphique du code ci-dessus

Création du fichier de style (« helloworld.css »)

| Code |
|--|
| <pre> /***** /* feuille de style */ /* helloworld.css */ /* Pierre Contri */ /* cree le 11/11/2005 */ /* mdf le 29/02/2008 */ *****/ @import "global/global.css"; @import "content/pictures.css"; /** corps */ BODY { color: #000000; font-family: sans-serif; font-size: 10pt; margin: 0px; background-color: black; } #divMenu { visibility: hidden; display: none; } /** Elements */ #divCorps { width: 911px; text-align: center; padding: 5px; margin: 10px auto 10px auto; background-color: white; } #idPage { height: 550px; overflow: auto; text-align: center; </pre> |

```
margin-top: 5px;
padding: 10px;
color: black;
background-color : #D5DDE3;
z-index: 1;
}

#divTitlePage {
color: #507896;
width: 911px;
height: 161px;
background-image: url(../../images/StMathieul.gif);
background-repeat: no-repeat;
text-align: center;
font-weight: bold;
font-size: 24pt;
font-style : italic;
z-index: 1;
}
```

Ce fichier doit être déposé dans l'emplacement : « ./sources/views/stylesheet ».

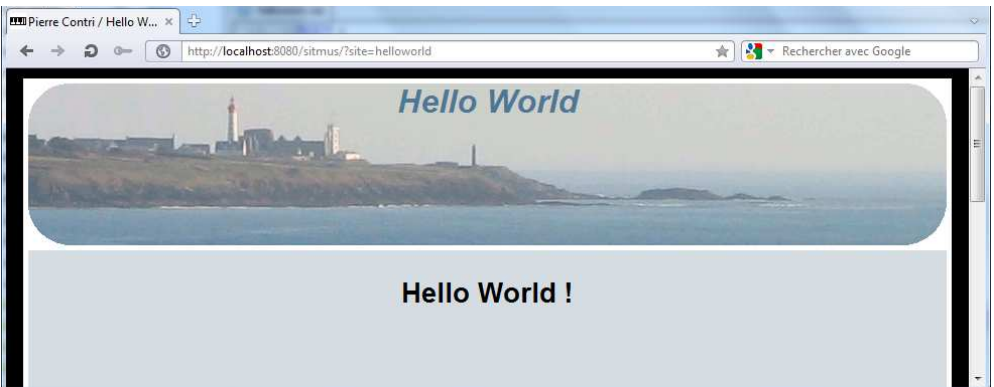


Illustration 16: Résultat graphique après application de la feuille de style

Annexes

Contenu de ce chapitre

| | |
|--|-----|
| Annexes..... | 75 |
| Code source de mon propre serveur Web basé sur PHP..... | 76 |
| Code source des fichiers Javascript..... | 83 |
| Gestion des fichiers « audio »..... | 83 |
| Gestion des fichiers « image »..... | 84 |
| Gestion du site..... | 93 |
| Gestion de l'attente entre l'appel client et le serveur..... | 97 |
| Code source des principales classes en PHP..... | 99 |
| Gestion des constantes avant l'entrée des objets..... | 99 |
| La classe abstraite « Documents »..... | 100 |
| La classe « PictureDocument »..... | 102 |
| La classe « MusicDocument »..... | 105 |
| La classe « TextDocument »..... | 105 |
| La classe « ToolDocument »..... | 106 |
| La classe « UnknownDocument »..... | 106 |
| La classe « DocumentsList »..... | 107 |
| La classe « FactoryDocument »..... | 110 |
| La classe « WebPage »..... | 112 |
| La classe « WebMasterPage »..... | 113 |

Code source de mon propre serveur Web basé sur PHP

```
<?php

error_reporting(E_ALL);

/* Allow the script to hang around waiting for connections. */
set_time_limit(0);

/* Turn on implicit output flushing so we see what we're getting
 * as it comes in. */
ob_implicit_flush();

server_loop("localhost", 8080);

$tabSessions = array();

/**
 * Creates a server socket and listens for incoming client
connections
 * @param string $address The address to listen on
 * @param int $port The port to listen on
 */
function server_loop($address, $port)
{
    global $tabSessions;

    $sock = socket_create(AF_INET, SOCK_STREAM, 0)
        or die("Failed to create socket !\n");

    $ret = socket_bind($sock, $address, $port)
        or die("Failed to bind socket !\n");

    $ret = socket_listen($sock, 0)
        or die("Failed to listen to socket !\n");

    print("Pending connection for clients\n");
```

```
while (true)
{
    $connection = @socket_accept($sock)
        or die("Error in socket acceptance !\n");

    $quiting = interact($connection);
    socket_close($connection);
    if($quiting) break;
}
socket_close($sock);
print("Bye !\n");
print_r($stabSessions);
}

function interact($socket)
{
    global $stabSessions;

    $InputData = "";
    $OutputStream = "";
    $stabReturn = array();
    $sid = "";

    $scanPrint = true;

    if(socket_recv($socket, $InputData, 4096, 0))
        print($InputData);
    $stabData = explode("\n", $InputData);

    $protocolRqt = "";
    $askRqt = "";
    $codeRqt = "HTTP/1.1";
    // get protocol, page and http version
    $stabRequest = explode(" ", $stabData[0]);
    if(count($stabRequest) > 0) $protocolRqt = $stabRequest[0];
    if(count($stabRequest) > 1) $askRqt = $stabRequest[1];
    if(count($stabRequest) > 2) $codeRqt = trim($stabRequest[2]);

    // get header information between browser and server
    $stabHeader = getHeader($stabData);
```

```
// default
if($askRqt == "/" ) $askRqt = "/index.html";

// split parametter and file name
$tabAskRqt = explode("?", $askRqt,2);
$pageName = $tabAskRqt[0];

// get url parameters
$_GET = array();
if(count($tabAskRqt) > 1)
    $_GET = explode("&",$tabAskRqt[1]);
if(count($_GET))
    splitKeyValue($_GET, '=');

// get post parameters
$_POST = array();
if($protocolRqt == "POST")
{
    $parameters = $tabData[count($tabData) - 1];
    $_POST = explode("&",$parameters);
    if(count($_POST))
        splitKeyValue($_POST, '=');
}
// merge the two parameters
$_REQUEST = array_merge($_GET, $_POST);

if(($protocolRqt == "GET" || $protocolRqt == "POST") &&
file_exists("." . $pageName))
{
    $tabReturn[] = $codeRqt . " 200 OK";
    $tabReturn[] = "Location: http://localhost:8080/";
    $tabReturn[] = date("r");
    $tabReturn[] = "Cache-Control: private";
    $tabReturn[] = "Server: CPierreWS";

    $expire = time() + 3600;
    // create or get SID (cookie & session),
    // create a cookie or use the existing one
    if(isset($tabHeader["Cookie"]))
    {
        $ssid = trim($tabHeader["Cookie"]);
        $tabReturn[] = "Cookie: " . $ssid;
```



```

    }
    else
    {
        $sid = "SID:ANON:localhost:" .
base64_encode(((double) rand()) * 1000000);
        $stabReturn[] = "Set-cookie: " . $sid;
    }

    $stabReturn[] = "WWW-Authenticate: Session,
realm=localhost";
    $stabReturn[] = "Allow: GET, HEADER, POST, PUT";

    if(strpos($pageName, ".php") !== false)
    {
        include(substr($pageName,1));
        $stabReturn[] = "Content-Type: text/html;
charset=UTF-8";
    }
    elseif(strpos($pageName, ".js") !== false ||
strpos($pageName, ".htm") !== false)
    {
        // include javascript or html page
        $OutputStream = implode(".", file("." .
$pageName));
        $stabReturn[] = "Content-Type: text/html;
charset=UTF-8";
    }
    elseif(strpos($pageName, ".jpg") !== false ||
strpos($pageName, ".bmp") !== false || strpos($pageName,
".ico") !== false || strpos($pageName, ".gif") !== false)
    {
        $imageFic = fopen("." . $pageName, "rb");
        if($imageFic)
        {
            $contents = "";
            while (!feof($imageFic))
                $contents .= fread($imageFic, 8192);
            fclose($imageFic);
        }
        $OutputStream = $contents;
        $stabReturn[] = "Content-Type: image/" .
substr($pageName, strlen($pageName) -3, 3);
        $scanPrint = false;
    }

```

```
    }

    $stabReturn[] = "Content-Length: " .
strlen($OutputStream);
    $stabReturn[] = "";
    $stabReturn[] = $OutputStream;
}
elseif(strtolower($InputData) == "bye" || $pageName ==
"/bye")
    $stabReturn[] = "Stop serveur : " . date("r");
elseif(!file_exists("." . $pageName))
    $stabReturn[] = "HTTP/1.0 404 Not Found";
else
    $stabReturn[] = "HTTP/1.0 400 Bad Request";

// concatain data array into a string
$returnTxt = implode("\n", $stabReturn);
if(socket_send($socket,$returnTxt,strlen($returnTxt),0))
{
    if($canPrint)
        print("\n{$returnTxt}\n");
    print("OK data returned\n\n\n");
}
else
    print("\nNOK data not returned\n\n\n");

// save datas into memory for the life time of running server
if($ssid != "") $stabSessions[$ssid] = $_REQUEST;

// return if ask to stop server
return(strtolower($InputData) == "bye" || strtolower($askRqt)
== "/bye");
}

function getHeader($stabSearch)
{
    if(count($stabSearch) < 1)
        return false;

    $tmpArr = array();
    reset($stabSearch);

    // skip the first line (protocole get, post, ..., parameters)
```

```
next($tabSearch);
// browse the next line until empty line
while($strTmp = next($tabSearch))
{
    if($strTmp == "")
        break;
    if(strpos($strTmp,":") !== false)
    {
        $tmpTab = explode(":", $strTmp, 2);
        $tmpArr[$tmpTab[0]] = trim($tmpTab[1]);
    }
}
return $tmpArr;
}

function splitKeyValue(&$tabToSplit, $splitCaract)
{
    if(count($tabToSplit) <= 0)
        return false;

    $tmpArr = array();
    foreach($tabToSplit as $value)
    {
        $tmpTab = explode($splitCaract, $value);
        if(count($tmpTab) > 1)
            $tmpArr[$tmpTab[0]] = urldecode($tmpTab[1]);
    }
    $tabToSplit = $tmpArr;
    return true;
}

?>
```

myMiniFramework for meta web site

Mini Framework pour site web basé sur PHP

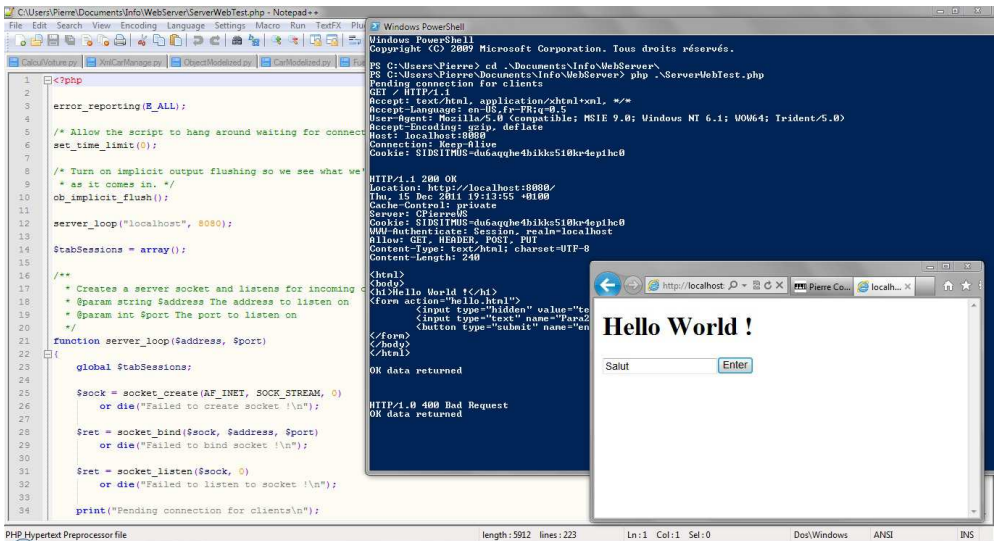


Illustration 17: Le serveur web en action

Code source des fichiers Javascript

Gestion des fichiers « audio »

```
function listen(fileSoundName,soundName,soundComment) {
    paramSafari = "width=300, height=130, toolbar=0, menubar=0,
scrollbars=3, resizable=1, status=0, location=0";
    paramNetscape = "width=400, height=180, toolbar=0, menubar=0,
scrollbars=3, resizable=1, status=0, location=0";

    if(navigator.appVersion.indexOf("Safari")>=0) {
        docMorceau = window.open(fileSoundName, "morceau",
paramSafari);
        docMorceau.focus();
    }
    else {
        docMorceau = window.open("about:blank", "morceau",
paramNetscape);
        docMorceau.document.writeln("<center><a
href=\"javascript:window.close();\">Fermer la
fen&ecirc;tre</a></center><br/>");
        docMorceau.document.write("<table width=\"100%\"><tr><td
width=\"100%\"><font color=\"black\"><u>" + soundName +
"</u></font></a><br>\n");
        docMorceau.document.write("</td></tr>\n");
        if(soundComment!=null && soundComment!="NULL")
        {
            docMorceau.document.write("<tr><td width=\"100%\"><font
color=\"black\">" + soundComment + "</font></td></tr>\n");
        }
        docMorceau.document.write("<tr><td width=\"100%\"><embed
name=\"PageMorceau\" src=\"\" + fileSoundName + "\" width=\"100%\"
height=\"30\" LOOP=FALSE AUTOSTART=TRUE CONTROLS=TRUE
MASTERSOUND /></td></tr>");
        docMorceau.document.write("<tr><td><a href=\"\" + fileSoundName
+ \"\">T&eacut;l&eacut;charger le morceau</a></td></tr>\n");
        docMorceau.document.write("</table>\n");
        docMorceau.focus();
    }
}
```

Gestion des fichiers « image »

```
TAILLE_HSCROLL = 40;
TAILLE_VSCROLL = 40;
MARGE_WITHOUT_SCROLL = 40;
TAILLE_SYSEXPLOIT = 100;

var isIE = (document.all)?1:0;

if(!Array.prototype.indexOf) {

    Array.prototype.indexOf = function(elmt /*, idx*/) {
        var len = this.length;
        var idx = Number(arguments[1]) || 0;
        idx = (idx < 0)?Math.ceil(idx):Math.floor(idx);
        if(idx < 0) idx += len;

        for(; idx < len; idx++) {
            if(idx in this && this[idx] === elmt)
                return idx;
        }
        return -1;
    };
}

function getSizeScreen() {
    return {x: ((isIE)?
document.body.offsetWidth:document.body.clientWidth),
        y: ((isIE)?
document.body.offsetHeight:document.body.clientHeight)};
}

function getSizeScreenDoc(docVar) {
    return {x: ((isIE)?docVar.offsetWidth:docVar.clientWidth),
        y: ((isIE)?docVar.offsetHeight:docVar.clientHeight)};
}

function getSizeInnerScreen() {
    var myWidth = 0, myHeight = 0;
    if( typeof( window.innerWidth ) == 'number' ) {
        //Non-IE
        myWidth = window.innerWidth;
```

```
    myHeight = window.innerHeight;
  } else if( document.documentElement &&
( document.documentElement.clientWidth ||
document.documentElement.clientHeight ) ) {
    //IE 6+ in 'standards compliant mode'
    myWidth = document.documentElement.clientWidth;
    myHeight = document.documentElement.clientHeight;
  } else if( document.body && ( document.body.clientWidth ||
document.body.clientHeight ) ) {
    //IE 4 compatible
    myWidth = document.body.clientWidth;
    myHeight = document.body.clientHeight;
  }
  return {x: myWidth, y: myHeight};
}

function getScrollXY() {
  var scrOfX = 0, scrOfY = 0;
  if( typeof( window.pageYOffset ) == 'number' ) {
    //Netscape compliant
    scrOfY = window.pageYOffset;
    scrOfX = window.pageXOffset;
  } else if( document.body && ( document.body.scrollLeft ||
document.body.scrollTop ) ) {
    //DOM compliant
    scrOfY = document.body.scrollTop;
    scrOfX = document.body.scrollLeft;
  } else if( document.documentElement &&
( document.documentElement.scrollLeft ||
document.documentElement.scrollTop ) ) {
    //IE6 standards compliant mode
    scrOfY = document.documentElement.scrollTop;
    scrOfX = document.documentElement.scrollLeft;
  }
  return {x: scrOfX, y: scrOfY};
}

function switchDiaporama() {
  var isDiapo = document.getElementById('isDiaporama');
  var arrayObj = document.getElementById('contentObjects');
  var divPict = document.getElementById('divPictureContent');
  var imgPict = document.getElementById('pictureImg');
  var buttonDiapo = document.getElementById('isDiapo');
```

```
if(!(isDiapo && arrayObj && divPict && buttonDiapo && imgPict))
    return false;

// change value in dynamic
isDiapo.value = (isDiapo.value == 1)?0:1;

// calcul for viewing
var valDiapo = (isDiapo.value == 1);
buttonDiapo.value = "Switch to the pictures " +
((valDiapo)?"table":"slideshow");

// check the picture containt
if(imgPict.alt == "empty") {
    load_picture('pictureImg', 'objElem', 'first');
}
divPict.style.visibility = (valDiapo)?'visible':'hidden';
divPict.style.display = (valDiapo)?'block':'none';

arrayObj.style.visibility = (valDiapo)?'hidden':'visible';
arrayObj.style.display = (valDiapo)?'none':'block';

return true;
}

function print_picture(nomPhoto) {

    var img = new Image();
    img.onload = function() {
        param = "top=0px, left=0px, width=" + screen.width + "px,
height=" + screen.height + "px, toolbar=0, menubar=0, scrollbars=3,
resizable=0, status=0, location=0"
        docPhoto = document.open("about:blank", "photo", param);

        var ratio = 1.0;
        var sizeScreen = getSizeScreen(docPhoto);
        var ratioX = (sizeScreen.x - TAILLE_VSCROLL) / this.width;
        var ratioY = sizeScreen.y / this.height;

        ratio = (ratioX < ratioY)?ratioX:ratioY;
        if(ratio > 1.0) ratio = 1.0;
        var pictureSizeX = Math.floor(ratio * this.width);
        var pictureSizeY = Math.floor(ratio * this.height);
```



```
        docPhoto.document.write('<html><head><title>Picture
Viewer</title></head><body style="background-color: black;">');
        docPhoto.document.write('<div style="margin: auto; text-align:
center; width: 100%; height: 100%;"><a
href="javascript:window.close()">');
        docPhoto.document.write('');
        docPhoto.document.write('<br /></a><br/>\n');

        docPhoto.document.write('</div></body></html>');
        docPhoto.focus();
    };
    img.src = nomPhoto;
    img.alt = nomPhoto;
}

function load_picture(nameImgContenuPhoto, nameInputListePhotos,
sens) {
    var imgContenuPhoto =
document.getElementById(nameImgContenuPhoto);
    var listPhotos =
document.getElementsByName(nameInputListePhotos);
    var nbPictures = listPhotos.length;

    if(imgContenuPhoto == null || listPhotos == null)
        return false;

    affichePatienter();

    // verifier qu'il y a des photos dans la liste
    if(nbPictures == 0) {
        alert("Pictures not found on the server");
        return false;
    }

    // recuperer la photo actuelle
    var srcPicture = imgContenuPhoto.src;
    // recuperer le nom de la photo uniquement si elle existe
    var pictureName = srcPicture.split("/").pop();
    if(pictureName.toLowerCase().indexOf(".jpg") == -1) {
        pictureName = "";
    }
}
```

```
    sens = "";
}

var pictureObj = null;

switch(sens) {
    case 'next' :
        pictureObj = getNextPicture(listPhotos, pictureName);
        break;
    case 'previous' :
        pictureObj = getPreviousPicture(listPhotos, pictureName);
        break;
    case 'random' :
        pictureObj = getRandomPicture(listPhotos);
        break;
    case 'first' :
        pictureObj = getNextPicture(listPhotos, null);
        break;
    case 'last' :
        pictureObj = getPreviousPicture(listPhotos, null);
        break;
    default :
        pictureObj = getNextPicture(listPhotos, null);
        break;
};

// recuperer son positionnement dans le tableau
var pictureInfo = pictureObj.value.split(';');
imgContenuPhoto.onload = function() {
    cachePatienter();
};
imgContenuPhoto.src = pictureInfo[0];
imgContenuPhoto.alt = (pictureInfo.length > 1) ? pictureInfo[1] :
pictureInfo[0];

updatePictureNumber(listPhotos, pictureInfo[0]);
return true;
}

function show_picture_fullScreen(pictureName, pictureAlt) {
    var divPicture      = "pictureImg";
    var divContentPict = "divPictureContent";
    var divTelecommand = "divPictureTelecommand";
```

```
affichePatienter();

var divPict = document.getElementById(divPicture);
var divCont = document.getElementById(divContentPict);
var divTel  = document.getElementById(divTelecommand);

// make div and image if it does not exists
if(divCont == null) {
    divCont = document.createElement('div');
    divCont.id = divContentPict;
}

if(divPict == null) {
    divPict = new Image();
    divPict.id = "pictureImg";
    divCont.appendChild(divPict);
}

if(isIE)
    divPict.onclick = function () { hide_div2(divContentPict); };
else
    divPict.addEventListener('click', function (e)
{ hide_div2(divContentPict); }, false);

// recuperation des dimensions du cadre
var sizeScreen = getSizeInnerScreen();
// remove scrollBar
sizeScreen.x -= 32;
// remove pb firefox
sizeScreen.y -= 34;

divCont.className = "fullScreenPicture";
document.body.appendChild(divCont);

show_div(divContentPict);
divPict.onload = function() {
    divPict.style.height = "auto";
    divPict.style.width = "auto";
    cachePatienter();
};
divPict.src = pictureName;
divPict.alt = pictureAlt;
```

```
// recalcul the page number of telecommand
if (divTel) {
    updatePictureNumber(document.getElementsByName('objElem'),
pictureName);
}
}

function hide_div2(divParent) {
    var idPage = document.getElementById('idPage');
    var objParent = document.getElementById(divParent);
    var divPict = document.getElementById('pictureImg');

    if(!idPage || !objParent || !divPict) return false;

    if(isIE)
        divPict.onclick = function ()
{ show_picture_fullScreen(divPict.src, divPict.alt); };
    else
        divPict.addEventListener('click', function (e)
{ show_picture_fullScreen(divPict.src, divPict.alt); }, false);

    objParent.className = "Picture";
    hide_div(divParent);

    // if is diaporama don't hide it, just move it
    var isDiapo = document.getElementById('isDiaporama');
    if(isDiapo) {
        idPage.appendChild(objParent);
        if(isDiapo.value == 1) show_div(divParent);
    }

    return true;
}

var defilementPhoto = false;
function defilStartStop(nameButtonDefil) {
    defilementPhoto = !defilementPhoto;
    var btnStartStop = document.getElementById(nameButtonDefil);
    if(btnStartStop != null)
        btnStartStop.className =
(defilementPhoto)? "buttonTelAutoStop": "buttonTelAutoStart";
}
```

```
function defilement_photo(nameImgContenuPhoto,
nameInputListePhotos, sens, nameInputTempo) {
    if(!defilementPhoto) return false;
    var inputTempo = document.getElementById(nameInputTempo);
    if(inputTempo != null) {
        var tempo = inputTempo.value * 1000;
        load_picture(nameImgContenuPhoto, nameInputListePhotos, sens);
        setTimeout("defilement_photo('" + nameImgContenuPhoto + "', '"
+ nameInputListePhotos + "', '" + sens + "', '" + nameInputTempo +
"')", tempo);
    }
    return true;
}

function updatePictureNumber(listByName, pictureName) {
    var pictureIdx = getObjectIndexInList(listByName, pictureName);
    if(pictureIdx == -1) return false;

    // afficher le repere de la photo photo xx / nbTotal
    var inputPageNb = document.getElementById('inputPageNb');
    if(inputPageNb)
        inputPageNb.value = (pictureIdx + 1) + " / " +
listByName.length;

    return true;
}

function getObjectIndexInList(listByName, searchName) {
    // get only the file name, without folders
    searchName = searchName.split('/').pop().toLowerCase();
    var idx = 0;
    // search
    while((idx < listByName.length) &&
(listByName[idx].value.toLowerCase().indexOf(searchName) == -1))
        idx++;
    // if not found, return -1
    return (idx >= listByName.length) ? -1 : idx;
}

function getNextPicture(pictureList, pictureName) {
    var idx = 0;
    // search the picture
```

```
    while((idx < pictureList.length) && (pictureList[idx+
+].value.indexOf(pictureName) == -1));
    // not found
    if(idx >= pictureList.length) idx = 0;
    return pictureList[idx];
}

function getPreviousPicture(pictureList, pictureName) {
    var idx = pictureList.length - 1;
    // search the picture
    while((idx >= 0) &&
(pictureList[idx--].value.indexOf(pictureName) == -1));
    // not found
    if(idx < 0) idx = pictureList.length - 1;
    return pictureList[idx];
}

function getRandomPicture(pictureList) {
    return pictureList[Math.round(Math.random() * (pictureList.length
- 1))];
}
```

Gestion du site

```
/*  
 * Gestion Site  
 * Pierre Contri  
 * le 12/2007  
 * mdf 04/2012  
 */  
  
if (typeof XMLHttpRequest == "undefined")  
    XMLHttpRequest = function () {  
        try { return new ActiveXObject("Msxml2.XMLHTTP.6.0"); }  
        catch (e) {}  
        try { return new ActiveXObject("Msxml2.XMLHTTP.3.0"); }  
        catch (e) {}  
        try { return new ActiveXObject("Microsoft.XMLHTTP"); }  
        catch (e) {}  
        //Microsoft.XMLHTTP points to Msxml2.XMLHTTP and is redundant  
        throw new Error("This browser does not support  
XMLHttpRequest.");  
    };  
  
var parseXml;  
  
if (typeof window.DOMParser != "undefined") {  
    parseXml = function(xmlStr) {  
        return ( new window.DOMParser() ).parseFromString(xmlStr,  
"text/xml");  
    };  
} else if (typeof window.ActiveXObject != "undefined" &&  
    new window.ActiveXObject("Microsoft.XMLDOM")) {  
    parseXml = function(xmlStr) {  
        var xmlDoc = new window.ActiveXObject("Microsoft.XMLDOM");  
        xmlDoc.async = "false";  
        xmlDoc.loadXML(xmlStr);  
        return xmlDoc;  
    };  
} else {  
    throw new Error("No XML parser found");  
}
```

```
var ie = document.all ? true : false;
var ns = document.layers ? true : false;

function refreshPage() {
    if(paciente != null)
        paciente();
    var formSite = document.getElementById('formSite');
    if(formSite)
        formSite.submit();
    else
        cachePatienter();
}

// obsolete, please, use goCategory
function goPage(categoryName) {
    if (arguments.length > 1)
        goCategory(categoryName, arguments[1]);
    else
        goCategory(categoryName);
}

function goCategory(categoryName) {
    if(paciente != null)
        paciente();

    var sitename = "";
    if (arguments.length > 1)
        sitename = arguments[1];

    var formSite = document.getElementById('formSite');
    if(formSite) {
        if (sitename != "")
            formSite.site.value = sitename;
        formSite.category.value = categoryName;
        formSite.submit();
    }
}

function goSite(sitepath) {
    goCategory('', sitepath);
}
```



```
function getCategoryWebS(categoryName) {
    var rqt = new XMLHttpRequest();
    if(!rqt) return false;

    // asynchronous answer
    rqt.onreadystatechange = function () {
        if(rqt.readyState == 4 && rqt.status == 200)
            document.getElementById('idPage').innerHTML =
rqt.responseText;
        else
            document.getElementById('idPage').innerHTML = rqt.readyState;
        cachePatienter();
    }

    // GET request
    //rqt.open('GET', 'http://localhost/sitmus/?getFunction=' +
categoryName, true);
    //rqt.send(null);

    // POST request
    rqt.open('POST', 'http://localhost/sitmus', true);
    rqt.setRequestHeader('Content-Type', 'application/x-www-form-
urlencoded');
    rqt.send("getFunction=" + categoryName);

    // synchronous answer
    //document.getElementById('idPage').innerHTML = rqt.responseText;
}

function show_div(divParent) {
    var obj = document.getElementById(divParent);
    if(!obj) return false;

    obj.style.visibility = 'visible';
    obj.style.display = 'block';

    return true;
}

function hide_div(divParent) {
    var obj = document.getElementById(divParent);
    if(!obj) return false;
```

```
obj.style.visibility = 'hidden';
obj.style.display = 'none';

return true;
}

function postMail() {
    var frmSite = (document.getElementById('formSite'))?
document.getElementById('formSite'):document.formSite;
    if(!frmSite) return false;

    frmSite.action = "sources/controls/sendmail.php";
    frmSite.target = "sendmail";
    frmSite.submit();
    frmSite.action = "index.php";
    frmSite.target = "_self";
    goPage("sitmus");

    return true;
}

function getAbsolutePosition(element) {
    var r = {x: element.offsetLeft, y: element.offsetTop};
    if(element.offsetParent) {
        var tmp = getAbsolutePosition(element.offsetParent);
        r.x += tmp.x;
        r.y += tmp.y;
    }
    return r;
}

function moveIcôneByMouse() {
    var icône = document.getElementById('icône');
    if(icône) {
        icône.style.top = event.y;
        icône.style.left = event.x;
    }
}
```

Gestion de l'attente entre l'appel client et le serveur

```
/* *****  
 * Gestion Patienter *  
 * Pierre Contri *  
 * le 14/12/2007 *  
 * ***** */  
  
var ie = (document.all) ? 1 : 0;  
var isIE6 = navigator.appVersion.indexOf("MSIE 6") > 0;  
var p = (ie) ? "filter" : "MozOpacity";  
var bAffichagePatienter = false;  
  
function patiente()  
{  
    affichePatienter();  
    var divp = document.getElementById('divPatienter');  
    if (divp != null) op(divp, 0);  
    progressOpacity('divPatienter', 50, 0, 70);  
}  
  
function progressOpacity(elementDivName, tpsTimeout, opacityStart,  
opacityEnd)  
{  
    var elementDiv = document.getElementById(elementDivName);  
    if(elementDiv == null) return false;  
    var testEnd = (ie) ? "alpha(opacity="+opacityEnd+)" :  
opacityEnd/100;  
    if (elementDiv.style[p] < testEnd)  
    {  
        op(elementDiv,opacityStart);  
        setTimeout("progressOpacity('" + elementDivName + "', " +  
tpsTimeout + ", " + eval(opacityStart + 5) + ", " + opacityEnd +  
")", tpsTimeout);  
    }  
}  
  
function op(n,v)  
{  
    v = (ie) ? "alpha(opacity=" + v + ")" : v / 100;  
    n.style[p] = v;  
}
```

```
function testAffichageDivPatienter()  
{  
    if (bAffichagePatienter)  
        cachePatienter();  
    setTimeout('testAffichageDivPatienter()', 2000);  
}  
  
function cachePatienter()  
{  
    var divpatientez = document.getElementById('divPatienter');  
    if (divpatientez != null)  
    {  
        divpatientez.style.visibility = 'hidden';  
        divpatientez.style.display = 'none';  
        bAffichagePatienter = false;  
    }  
}  
  
function affichePatienter()  
{  
    var divpatientez = document.getElementById('divPatienter');  
    if (divpatientez == null)  
        divpatientez = createPatienter();  
  
    divpatientez.style.visibility = 'visible';  
    divpatientez.style.display = 'block';  
    bAffichagePatienter = true;  
}  
  
function createPatienter()  
{  
    var divpatientez = document.createElement('divPatienter');  
    divpatientez.setAttribute("id", "divPatienter");  
    divpatientez.setAttribute("name", "divPatienter");  
    var contentPatienter = "<div id=\"divPatienterIntern\"><img  
src=\"../sources/images/hourglass2.gif\" alt=\"patiente\"/><br /><p  
id=\"textPatienter\">Merci de patienter pendant le chargement de la  
page</p></div>";  
    divpatientez.innerHTML = contentPatienter;  
    document.body.appendChild(divpatientez);  
    return divpatientez;  
}
```

Code source des principales classes en PHP

Gestion des constantes avant l'entrée des objets

```
/**
 * Constantes Definition
 * About : - documents type
 *         - object structure for data (in case of object)
 *         - array extension type
 */

/*
define('DOCTYP_PICTURE','PictureDocument');
define('DOCTYP_MUSIC','MusicDocument');
define('DOCTYP_TEXT','TextDocument');
define('DOCTYP_TOOL','ToolDocument');
define('DOCTYP_DOCUMENTSLIST','DocumentsList');
define('DOCTYP_UNKNOWN','UnknownDocument');

define('OBJDATA_ID','id');
define('OBJDATA_TITLE','title');
define('OBJDATA_PATH','path');
define('OBJDATA_SUBJECT','subject');
define('OBJDATA_COMMENTS','comments');
define('OBJDATA_TYPE','filetype');

define('TYPEXT_PICTURE','jpg,bmp,gif');
define('TYPEXT_MUSIC','mp3,wav,wma');
define('TYPEXT_DOCUMENT','doc,pdf,ppt,pptx,pps,docx,odt,txt');
define('TYPEXT_TOOL','zip,7z');
*/
```

La classe abstraite « Documents »

```
abstract class Document {
    var $id = 0;
    var $title = "";
    var $subject = "";
    var $path = "";
    var $comments = "";

    static $type = null;

    public function __construct($pathfile = "") {
        $this->path = $pathfile;
    }

    public function tryParseOnItself($enr, $dirName = ".") {
        $idx = 0;
        $this->id      = trim($enr[$idx++]);
        $this->title    = (count($enr) > 4)?trim($enr[$idx++]):"";
        $this->path     = $dirName . "/" . trim($enr[$idx++]);
        $this->comments = trim($enr[$idx++]);
        $this->subject  = trim($enr[$idx++]);
    }

    public function __toString() {
        $tabReturn = array();
        $reflector = new ReflectionClass($this);
        $docProperties = $reflector->getProperties();
        foreach($docProperties as $docProp) {
            if($docProp->getValue($this) != "")
                $tabReturn []= ucfirst($docProp->getName()) . ": " .
                    $docProp->getValue($this);
        }
        return implode("<br />\n", $tabReturn);
    }

    public function getObjectHiddenField() {
        return <<<ENDObj
<input type="hidden" name="objElem" id="objElem{$this->id}"
value="{ $this->path }; { $this->comments }" />
ENDObj;
    }
}
```

```

function checkAndCorrectPath() {
    $infoFile = pathinfo($this->path);
    $nomFileLower = $infoFile['dirname'] . "/" .
strtolower($infoFile['basename']);
    $nomFileUpper = $infoFile['dirname'] . "/" .
strtoupper($infoFile['basename']);

    if(file_exists($this->path));
    elseif(file_exists($nomFileLower))
        $this->path = $nomFileLower;
    elseif(file_exists($nomFileUpper))
        $this->path = $nomFileUpper;
    else
        return false;

    return true;
}

public function printObj($increment = 4) {
    if(!$this->checkAndCorrectPath())
        return "File '{$this->path}' does not exist";
    return "";
}

/* only in php 5.3 */
/*
public static function isDocumentType($filepath) {
    $tmpArr = null;
    $extension = "";

    try {
        $tmpArr = explode(".", $filepath);
        $extension = strtolower(array_pop($tmpArr));
    } catch(Exception $iep) {
        return false;
    }

    return (strpos(static::$typeext, $extension) !== false);
}
*/
}

```

La classe « *PictureDocument* »

```
class PictureDocument extends Document {

    public static $typext = 'jpg,bmp,gif';

    public function printObj($increment = 4) {
        parent::printObj($increment);
        $strIncrement = str_pad("", $increment, " ", STR_PAD_LEFT);

        // get compress picture if it exists
        $infoFile = pathinfo($this->path);
        $file2 = $infoFile['dirname'] . "/compress/cmp_" .
        $infoFile['basename'];
        $file_cmp = (file_exists($file2)) ? $file2 : $this->path;

        $strObj = $strIncrement . "<div class=\"" . get_class($this) .
        "\">\n";
        $strObj .= $strIncrement . " <img id=\"pictureImg{$this->id}\"
        src=\"{$file_cmp}\" class=\"pictureImgTab\" alt=\"{$this-
        >comments}\" onclick=\"javascript:show_picture_fullScreen('{$this-
        >path}','{$this->comments}')\"
        onmouseover=\"this.style.cursor='hand';\" /><br/>\n";

        if($this->comments != "")
            $strObj .= $strIncrement . " <div class=\"libele\">{$this-
            >comments}</div>\n";
        $strObj .= $strIncrement . "</div>\n";

        return $strObj;
    }
}
```



```

public static function printPictureManagement($sourcesList) {

    $isDiaporama = false;
    $isDiapoJavaScript = (int)$isDiaporama;
    $translateSlideShow = ($isDiaporama)?"table":"slideshow";
    $showInSlideShow = (!$isDiaporama)?"style=\"visibility: hidden;
display: none;\"":"";

    return <<<ENDPict

        <!-- Picture management part -->
        <div id="divDiapoManage">
            <input type="hidden" id="isDiaporama" name="isDiaporama"
value="{ $isDiapoJavaScript}" />
            <input type="button" id="isDiapo" class="btnsubmit"
value="Switch to the pictures { $translateSlideShow}"
onclick="javascript:switchDiaporama();" />
            <br />
        </div> <!-- divDiapoManage -->

        <!-- Picture Documents -->
        <div id="divPictureContent" class="PictureDocument"
{ $showInSlideShow}
onmouseover="javascript:show_div('divPictureTelecommand');"
onmouseout="javascript:hide_div('divPictureTelecommand');">
            

            <div id="divPictureTelecommand"
class="divPictureTelecommand">
                <input type="button" id="bPreviousPicture"
class="buttonTel" value=" "
onclick="javascript:load_picture('pictureImg', 'objElem',
'previous');" onmouseover="this.style.cursor='hand';" />
                <input type="button" id="bRandomPicture"
class="buttonTel" value=" "
onclick="javascript:load_picture('pictureImg', 'objElem',
'random');" onmouseover="this.style.cursor='hand';" />
                <input type="button" id="bNextPicture" class="buttonTel"
value=" " onclick="javascript:load_picture('pictureImg', 'objElem',
'next');" onmouseover="this.style.cursor='hand';" />

```

```

        <input type="text" id="txtTimeOut" class="inputTel"
size="1" maxlength="2" value="05" />
        <input type="button" id="bDelimentAuto"
class="buttonTelAutoStart" value=" "
onclick="javascript:defilStartStop(this.id);defilement_photo('pictu
reImg', 'objElem', 'next', 'txtTimeOut');"
onmouseover="this.style.cursor='hand';" />
        <input type="text" id="inputPageNb" class="inputTel"
size="8" value="0" readonly="readonly" />

        <!-- Objects source list -->
        {$sourcesList}
        <!-- <script
type="text/javascript">load_picture('pictureImg', 'objElem',
'random');</script> -->
        </div> <!-- divPictureTelecommand -->

</div> <!-- divPictureContent -->
ENDPict;
    }
}
```

La classe « *MusicDocument* »

```
class MusicDocument extends Document {

    public static $typext = 'mp3,wav,wma';

    public function printObj($increment = 4) {
        parent::printObj($increment);
        $strIncrement = str_pad("", $increment, " ", STR_PAD_LEFT);
        $strObj = $strIncrement . "<div class=\"" . get_class($this) .
"\">>\n";
        $strObj .= $strIncrement . "  <a href=\"javascript: return
false;\" onclick=\"javascript:listen('{ $this->path}', '{ $this-
>title}', '{ $this->comments}');\">{ $this->title}</a><br />\n";
        if($this->comments != "" && $this->comments != "NULL")
            $strObj .= $strIncrement . " " . $this->comments . "\n";
        $strObj .= $strIncrement . "</div>\n";

        return $strObj;
    }
}
```

La classe « *TextDocument* »

```
class TextDocument extends Document {

    public static $typext = 'doc,pdf,ppt,pptx,ppsx,pps,docx,odt,txt';

    public function printObj($increment = 4) {
        parent::printObj($increment);
        $strIncrement = str_pad("", $increment, " ", STR_PAD_LEFT);
        return $strIncrement . "<a href=\".\" \"
onclick=\"javascript:window.open(' { $this->path} ');return
false;\">{ $this->title}</a>\"";
    }
}
```

La classe « ToolDocument »

```
class ToolDocument extends Document {  
  
    public static $typext = 'zip,7z,tar,gz';  
  
    public function printObj($increment = 4) {  
        parent::printObj($increment);  
        $strIncrement = str_pad("", $increment, " ", STR_PAD_LEFT);  
        return $strIncrement . "<a href=\".\\\"  
onclick=\"javascript:window.open('{ $this->path}'); return  
false;\">{ $this->title}</a>\";  
    }  
}
```

La classe « UnknownDocument »

```
class UnknownDocument extends Document {  
  
    public static $typext = '';  
  
    public function printObj($increment = 4) {  
        parent::printObj($increment);  
        $strIncrement = str_pad("", $increment, " ", STR_PAD_LEFT);  
  
        return $strIncrement . (string)$this;  
    }  
}
```

La classe « DocumentsList »

```
class DocumentsList extends ArrayObject {
    var $subject = "";
    var $childrenDocumentsList = array();

    public function __construct($array = array()) {
        if(count($array))
            $this->importArray($tmpArray);
    }

    public function checkAndCorrectPath() {
        return true;
    }

    public function importArray($tmpArray) {
        foreach($tmpArray as $tmpEntry)
            $this []= $tmpEntry;
    }

    public function printObj($recurse = true, $increment = 4) {
        $strDoc = "";
        $strIncrement = str_pad("", $increment, " ", STR_PAD_LEFT);
        $pInfo = pathinfo($this->subject);
        $tName = str_replace("_", " ", ucfirst($pInfo['basename']));
        $strDoc .= $strIncrement . "<div style=\"text-align:
left;\">\n";
        $strDoc .= $strIncrement . "<h3>{$tName}</h3>\n";
        $strDoc .= $strIncrement . " <ul>\n";
        foreach($this as $f) {
            $strDoc .= $strIncrement . " <li>" . $f->
printObj($increment) . "</li>\n";
        }
        $strDoc .= $strIncrement . " </ul>\n" . $strIncrement .
"</div>\n";

        if($recurse)
            foreach($this->childrenDocumentsList as $f)
                if(is_a($f, 'TextDocument') && $f->subject != "") $strDoc .=
$f->printObj($increment);
    }
}
```

```
        return $strDoc;
    }

    public function fillContentObjects($increment = 4) {
        $strIncrement = str_pad("", $increment, " ", STR_PAD_LEFT);

        if(!count($this) && count($this->childrenDocumentsList)) {
            $this->importArray($this->childrenDocumentsList);
            $this->childrenDocumentsList = array();
        }

        if(!count($this))
            return "<p>There is no content to print</p>\n";

        $strContent = "";

        /* Picture Management */
        if(is_object($this) && $this-
>isArrayContaintMultimediaCategory('PictureDocument') &&
count($this) > 1) {
            $strContent .= PictureDocument::printPictureManagement($this-
>getObjectsHiddenFields());
        }

        // filling array
        $strContent .= $strIncrement . "<div id=\"contentObjects\">\n";
        foreach($this as $enr) {
            $strContent .= $strIncrement . "    <div
class=\"ContentObj\">\n";
            $strContent .= $enr->printObj($increment + 4);
            $strContent .= $strIncrement . "    </div> <!-- ContentObj
-->\n";
        }
        $strContent .= $strIncrement . "</div> <!-- contentObjects
-->\n";

        return $strContent;
    }
}
```

```
public function getObjectsHiddenFields($separator = "") {
    $inputObjectsList = "";
    foreach($this as $obj) {
        $inputObjectsList .= $obj->getObjectHiddenField() .
$separator;
    }
    return $inputObjectsList;
}

public function isArrayContaintMultimediaCategory($searchType) {
    foreach($this as $tmpElem) {
        if(is_a($tmpElem, $searchType))
            return true;
    }
    return false;
}
}
```

La classe « FactoryDocument »

```
class FactoryDocument {
    public static $typeDocumentsList = array('PictureDocument',
'MusicDocument', 'TextDocument', 'ToolDocument',
'UnknownDocument');

    public static $instance = null;

    private function __construct() {
    }

    public static function getInstance() {
        if(self::$instance == null) {
            $className = __CLASS__;
            self::$instance = new $className;
        }
        return self::$instance;
    }

    public static function createDocument($enr) {
        $doc = null;
        $tmpPath = "";
        try {
            // if title is present, switch to new column n°2, else n°1
            if(is_array($enr))
                $tmpPath = trim($enr[(count($enr) > 4)?2:1]);
            else
                $tmpPath = $enr;

            $docType = FactoryDocument::getMultimediaType($tmpPath);
            $doc = new $docType($tmpPath);
        } catch (Exception $exp) {
            $doc = new UnknowDocument();
        }
        return $doc;
    }
}
```



```
public static function getMultimediaType($filepath) {

    /* php 5.1 */
    $tmpArr = null;
    $extension = "";

    try {
        $tmpArr = explode(".", $filepath);
        $extension = strtolower(array_pop($tmpArr));
    } catch(Exception $iep) {
        return false;
    }

    if(strpos(PictureDocument::$typeext, $extension) !== false)
return 'PictureDocument';
    else if(strpos(MusicDocument::$typeext, $extension) !== false)
return 'MusicDocument';
    else if(strpos(TextDocument::$typeext, $extension) !== false)
return 'TextDocument';
    else if(strpos(ToolDocument::$typeext, $extension) !== false)
return 'ToolDocument';
    else return 'UnknownDocument';

    /* php 5.3 */
    /* ask to each documents if the file is anything of it */
    /*
    foreach(FactoryDocument::$typeDocumentsList as $tmpDocType) {
        if($tmpDocType::isDocumentType($filepath)) {
            return $tmpDocType;
        }
    }
    return 'UnknownDocument';
    */
}
```

La classe « WebPage »

```
class WebPage {
    var $docTitle      = "";
    var $metaWords     = "";
    var $contentPage = "";

    public function getPageArray() {
        return array("docTitle" => $this->docTitle, "contentpage" =>
$this->contentPage, "metaWords" => $this->metaWords);
    }

    public function __toString() {
        return "Name : {$this->name} / Title : {$this->docTitle} /
ContentPage : {$this->contentPage} / MetaWords : {$this-
>metaWords}";
    }
}
```

La classe « *WebMasterPage* »

```

class WebMasterPage extends WebPage {
    private $pageTitle    = "";
    public  $category      = "";
    public  $styleFile     = "";
    public  $webSiteName   = "";
    public  $javascripts   = "";
    public  $menu          = null;

    public function __construct($wpCategory = "", $cpyTempPage =
null) {
        if($wpCategory != "")
            $this->setWebPage($wpCategory, $cpyTempPage);
    }

    public function __get($attr) {
        if($attr == "pageTitle")
            return $this->menu->getEntryByName($this->category)-
>titlePage;
        else if(isset($this->$attr)) return $this->$attr;
        else throw new Exception('Unknow attribute '.$attr_pageTitle);
    }

    public function setWebPage($wpCategory, $newWebPage) {
        $this->category = $wpCategory;
        $this->importWebPage($newWebPage);
    }
    public function importWebPage($cpyWebPage) {
        if ($cpyWebPage != null && is_a($cpyWebPage, 'WebPage')) {
            $this->docTitle    = $cpyWebPage->docTitle;
            $this->metaWords   = $cpyWebPage->metaWords;
            $this->contentPage = $cpyWebPage->contentPage;
        }
    }
    public function getMenuHTML($increment = 8) {
        if ($this->menu == null) return "";
        return $this->menu->printCategory($this->category, $increment);
    }
}

```

Bibliographie

- PHP : « PHP 5 » 3e édition, L. Atkinson, CampusPress
- Python : « Python », M. Brucher, ENI
- JavaScript : « JavaScript » La référence, D. Flanagan, O'Reilly
- CSS : « CSS Avancées vers HTML5 et CSS3 », R. Goetter, Eyrolles
- Design Patern : « Design Patterns pour C# », L. Debrauwer, ENI
- HTML 4 : « HTML 4 XML-XHTML », R. Steyer, Micro Application
- DotNet : « .Net 2.0 Framework » 70-536, Microsoft® Press
- UML : « UML 2 », L. Debrauwer et F. Van Der Heyde, ENI

Sites Web de référence

- « <http://www.w3.org> »
- « <http://www.php.net> »
- « <http://www.alsacreations.com> »

Site Web de l'auteur

- « <http://pierre.contri.free.fr> »
- « <http://pierre.contri.free.fr/?site=programming&category=myFramework> »



Illustration 18: Visitez le site de l'auteur

Contact

« pierre.contri@free.fr »

Consultant Technique

pierre.contri@free.fr



FORMATION

2012 : Administration Windows 2008 par la programmation Powershell et Active Directory
2011 : Administration Oracle 11g
2010 : Management ADAM
2009 : Développement de bases de données sous Lotus
2008 : Certification blanche MCP : .Net Bases Framework (70-536)
2007 : Certification MCP : Microsoft SQL Server 2005 (70-431)
2007 : Présentation de Sharepoint 2007
2005 : Ingénieur ENIB option Génie Logiciel
2002 : Brevet de Technicien Supérieur en Informatique Industrielle
1999 : Baccalauréat STI génie électrotechnique

EXPERIENCES

2012 : Analyste-développeur Logica
2008 : Chef de Projet Technique ThyssenKrupp Presta France
2005 : Ingénieur d'études Ariane-Ingénierie

SITE

« <http://pierre.contri.free.fr> »



L'auteur est un passionné de musique et d'informatique.

A l'ENIB il a obtenu son diplôme d'ingénieur en informatique industrielle option génie logiciel et travaille comme tel depuis.

En dehors de son travail il s'est lancé dans la création de son propre site et en a créé pour ses amis. Par plaisir de programmer ou par nécessité (vu la hauteur de ses exigences) il a développé lui-même son propre Framework.

Son site a évolué au cours du temps selon ses envies ou ses besoins.

Sa curiosité et sa quête permanente d'améliorations l'ont porté à imaginer un site permettant de loger d'autres sites.

Cette particularité alliée à la conception évolutive des différentes fonctions du site confère à ce Framework une potentialité tout à fait remarquable.

A exploiter par des entreprises, des associations ou simplement par des réseaux d'amis...
Et il se fera un plaisir d'offrir le fichier à qui est intéressé !

Montigny, septembre 2012

B.C.