

PHP 5 avancé

6^e édition

Éric Daspét

Cyril Pierre de Geyer

Avec la contribution de Frédéric Hardy

Préface de Damien Seguy



EYROLLES

© Groupe Eyrolles, 2012, ISBN : 978-2-212-13435-3

Table des matières

Avant-propos	XXXIII
Pourquoi ce livre ?	XXXIII
Structure de l'ouvrage	XXXIII
Remerciements	XXXVI
CHAPITRE 1	
Qu'est-ce que PHP ?	1
Introduction à PHP	1
Un langage Open Source	1
Que faire avec PHP ?	3
Particularités de PHP	4
Historique	6
Mode de développement du projet PHP	9
Principaux apports de PHP 5	10
Les principaux apports des versions 5 à 5.2 de PHP	10
Les principaux apports de PHP 5.3	11
Architecture et fonctionnement	12
Architecture technique	12
Fonctionnement de PHP	13
PHP en France et dans le monde	15
La communauté française	16
Les ressources d'aide francophones	16
Les ressources d'aide anglophones	22

CHAPITRE 2

Installer et configurer	25
Migration vers les dernières versions de PHP	25
Modes d'installation	26
CGI, FastCGI et FPM	26
Modules	26
Installer PHP 5 sous MS-Windows	26
Installation automatique	26
Installation manuelle	27
Installer PHP 5 sous Unix	35
Installation automatisée	35
Installation manuelle d'Apache	36
Installation manuelle de MySQL	37
Installation manuelle de PHP	38
Gestion des droits d'accès	41
Modules additionnels PECL	41
Configuration de PHP avec php.ini	41
Utilisation des modules et des extensions	42
Les directives de configuration	43
Gestion de la configuration	47
Utiliser PHP-FPM	50
Installation et compilation	50
Gestion de la configuration	50
Intégration avec un serveur web	53

CHAPITRE 3

Les structures de base	55
Insertion de PHP dans HTML	55
Balises d'ouverture et de fermeture	56
Les commentaires	56
Enchaînement des instructions	57
Structure du document	58
Exécuter du code PHP	59
Constantes et variables	61
Variables	61
Constantes	65

Types de données	67
Booléens (boolean)	67
Les nombres entiers (integer)	68
Les nombres flottants (double, float)	68
Les chaînes de caractères (string)	68
Les tableaux (array)	74
Transtypage	78
 CHAPITRE 4	
Traitements de base	81
Les opérateurs	81
Opérateurs d'affectation	81
Opérateurs arithmétiques	83
Opérateurs combinés	86
La concaténation	86
Opérateurs de comparaison	87
Opérateurs logiques	88
Opérateurs sur les bits	89
Opérateur ternaire	89
Priorités entre opérateurs	90
Structures de contrôle	90
Les conditions	91
Les boucles	96
Les instructions d'arrêt	101
L'instruction goto	102
Les fonctions utilisateur	102
Déclaration d'une fonction	103
Appel de fonction	104
Visibilité des variables	104
Retourner plusieurs valeurs	107
Nombre de paramètres indéfini	108
Fonctions anonymes	108
Fermetures lexicales	109
Inclure des fichiers	110
Différence entre require() et include()	112
require_once() et include_once()	112

CHAPITRE 5

Traitements de chaînes	113
Fonctions d’affichage	113
Affichages simples	113
Affichages avec masques	114
Informations sur une chaîne	117
Accéder à un caractère précis	117
Valeur ASCII d’un caractère	117
Taille d’une chaîne	118
Position d’une sous-chaîne	119
Présence de certains caractères	120
Conversions et formatages	120
Protections et échappements	120
Conventions d’affichage locales	125
Jeux de caractères	126
Manipulations sur les chaînes	129
Recherche d’une sous-chaîne	130
Récupérer une sous-chaîne	130
Remplacer un motif	130
Fonctions d’élagage	131
Remplissage	132
Changement de casse	132
Coupure de paragraphes	133

CHAPITRE 6

Utilisation des tableaux	135
Déclaration et initialisation d’un tableau	135
Déboguer un tableau	136
Taille d’un tableau	136
Tester l’existence et le type d’un tableau	137
Recherche d’un élément	138
Présence dans le tableau	138
Recherche de la clé correspondante	139
Nombre d’occurrences d’un élément	140
Récupération aléatoire d’éléments	140

Trier les tableaux	141
Tri par valeur	141
Tri en ordre inverse	141
Garder les associations clé-valeur	142
Tri par clé	142
Tri naturel	142
Trier avec une fonction utilisateur	143
Tri multicritère	144
Extractions et remplacement	145
Affecter des variables	145
Sérialisation de tableaux	145
Extraction d'un sous-tableau	146
Remplacement d'un sous-tableau	146
Gestion des clés et des valeurs	147
Liste des clés utilisées	147
Liste des valeurs utilisées	148
Échanger les clés et les valeurs	148
Fusions et séparations	148
Fusion de plusieurs tableaux	148
Séparation d'un tableau en plusieurs	150
Différences et intersections	151
Différences entre tableaux	151
Intersections entre deux tableaux	151
Gestion des doublons	152
Gestion des piles et des files	152
Navigation dans les tableaux	153
 CHAPITRE 7	
Fonctions usuelles	155
Fonction d'affichage	155
Informations de configuration	155
Affichage de débogage	158
Coloration syntaxique de code	159
Fonctions mathématiques	160
Connaître les extrémités	160
Arrondir des valeurs	160

Créer des valeurs aléatoires	161
Travailler sur différentes bases	162
Fonctions de date	163
Formater une date/heure locale	163
Fonctions réseau	168
Résolution DNS d'une adresse IP	168
Corrélation IP/DNS	169
Fonctions de chiffrement	171
Quelques définitions : chiffrement, hachage, codage/décodage	171
Fonctions de hachage	171
Fonctions de codage et décodage	176
Exécution de code	176
Fonction à l'arrêt du script	176
Exécution d'une chaîne de code PHP	177
Cas d'application	178
Login/mot de passe sécurisés	178

CHAPITRE 8

Formulaires et superglobales	181
Formulaires HTML	181
Nouveautés depuis PHP 4.0	182
Caractères spéciaux et HTML	182
Création du formulaire	183
Déclaration d'un formulaire	183
Méthode d'envoi du formulaire	183
Envoi d'images et de fichiers	185
Réception des données en PHP	185
Utilisation des superglobales	185
Récupération d'une donnée simple	186
Retours à la ligne et zones de texte	188
Utilisation des cases à cocher	189
Validation de données avec l'extension Filter	190
Listes à sélections multiples	196
Gestion des images cliquables	197
Téléchargements d'images et de fichiers	198
Formulaire dynamique et tableaux	200

Autres problématiques	201
Gestion du temps	201
Gestion de la taille des données	201
Stockage des fichiers temporaires	201
Sécurité et données reçues	202
Procédure de gestion des formulaires	202
CHAPITRE 9	
Environnement web et superglobales	205
Descriptif du contexte web	205
Client-serveur	205
En-tête et contenu	206
Variables superglobales	207
Informations sur le serveur	208
Nom du serveur	208
Racine du serveur	208
Autres informations sur le serveur	209
Authentification HTTP	209
Principes du protocole HTTP	210
Gestion avec PHP	211
Authentification par le serveur web	212
Paramètres de la connexion	213
Adresse IP et port du client	213
Adresse IP et port du serveur	214
Description de la requête HTTP	215
Paramètres de la requête	215
L'adresse demandée (URL)	216
Informations fournies par le client	217
Environnement système	219
Nom du script exécuté	219
Interactions PHP/JavaScript	219
Ligne de commande	220
Lecture des arguments	220
Nombre d'arguments	221

CHAPITRE 10

Les cookies	223
Présentation	223
Forme du cookie sur votre ordinateur	224
Lecture et écriture d'un cookie	225
Envoi d'un cookie	225
Lecture d'un cookie	226
Suppression d'un cookie	228
Modifier les valeurs d'un cookie	228
Validité et date d'expiration	229
Tableaux et types complexes	230
Restriction de portée du cookie	231
Limitations et sécurité	233
Limitations dues aux navigateurs	233
Les cookies n'ont aucune sécurité	233
Cas d'application	234
Outil de personnalisation d'affichage	234

CHAPITRE 11

Les sessions	239
Qu'est-ce qu'une session ?	239
Lecture et écriture	240
Utilisation avancée	241
Fonctionnement interne des sessions	242
Suppression d'une session	243
Définition manuelle de l'initialisation	243
Stockage des données de session	244
Paramètres du cookie	245
Accès concurrents aux sessions	245
Configuration de PHP	246
Initialisation des sessions	246
Stockage des données de session	246
Paramètres du cookie	247
Expiration des sessions	247
Gestion du cache	247
Transmission de l'identifiant	248

Gestionnaires de sessions	249
Définir un gestionnaire personnalisé	250
Limitations et sécurité	253
Cachez les sessions	253
N'utilisez pas la réécriture des liens	253
Les identifiants par défaut suffisent	253
Attaque par fixation de session	254
Vérifiez l'identité de l'utilisateur	254
N'ayez pas confiance	254
Cas d'application	255
Authentification par formulaire	255
CHAPITRE 12	
Gestion des objets	261
Introduction aux objets	261
Pourquoi programmer en objet ?	261
Qu'est-ce qu'un objet ?	262
Qu'est-ce qu'une classe ?	262
Qu'est-ce qu'une instance ?	263
Utilisation simple des objets	263
Déclarer une classe	263
Utilisation des objets	265
Vérifier le type d'un objet	270
Copie et référence	271
Le comportement PHP 4	272
PHP 5, le passage par référence	273
Garder la compatibilité avec PHP 4	274
La copie explicite d'objet, ou clonage	274
Égalité et identité	276
Constructeurs et destructeurs	277
Constructeur	277
Destructeur	279
La notion d'héritage	280
Définition de la notion d'héritage	280
Définition d'une classe héritée	280
Redéfinition d'attribut ou de méthode	281
Accès aux méthodes parentes	283

Sûreté de programmation	283
Contrôle d'accès	284
Typage	287
Classes abstraites et interfaces	288
Accès statiques	292
Accès à une classe arbitraire	292
Définition en vue d'un accès statique	293
Accès à la classe en cours	293
Accès à la classe parente	294
Résolution statique à la volée	294
Les espaces de noms	295
Définition d'un espace de noms	296
Plusieurs espaces de noms dans un même fichier	297
Espace de noms racine	298
Imbrication d'espaces de noms	298
Utilisation d'un élément d'un espace de noms	299
Utilisation d'un nom absolu	300
Cas de la programmation dynamique	301
Déterminer l'espace de noms courant	301
Importer des espaces de noms	302
Alias d'espace de noms	304
Conventions de nommage	305
Chargement automatique des classes (autoload)	306
Définition d'une fonction de chargement automatique	307
Fonction de chargement automatique native	308
Supprimer une fonction de chargement automatique	308
Obtenir la liste des fonctions de chargement automatique	308
Forcer la définition d'une classe via le chargement automatique	309
La fonction <code>__autoload()</code>	309
Utilisation via les sessions	310
Utilisation de <code>__sleep()</code> et <code>__wakeup()</code>	310
L'interface <code>Serializable</code>	310
Surcharge	311
Affectations des attributs	312
Lecture d'attribut (Mutator)	312
Test d'existence d'un attribut	313
Suppression d'un attribut	313

Appel d'une méthode (Accessor)	313
Appel d'une méthode statique	313
Itérateurs	314
Utilisation simple	314
Utilisation complète	317
Notations d'index	318
Coupler PHP et UML	319
Introspection	320
Principes pour démarrer	321
Les fonctions	322
Les objets, classes et interfaces	326
Les attributs	328
CHAPITRE 13	
Gestion de fichiers	329
Lecture et écriture	329
Fonctions d'accès rapide	330
Ouverture d'un fichier	336
Lecture d'un fichier	339
Écriture dans un fichier	340
Positions dans le fichier	341
Détection de fin de fichier	343
Fermeture d'un fichier	343
Gestion du tampon	343
Accès concurrents	344
Manipulation de fichiers	347
Copie et déplacement	347
Création et effacement	347
Liens	348
Gestion des répertoires	349
Parcourir un répertoire	350
Position dans l'arborescence	352
Créations et effacements	353
Informations sur les fichiers	353
Existence d'un fichier	354
Dates de fichiers	354

Taille de fichier	355
Espace disque disponible	355
Nom et adresse d'un fichier	356
Nature des fichiers	356
Liens symboliques	357
Permissions et droits d'accès	357
Changement de propriétaire	359
Modifier les permissions.	359
Masque par défaut.	359
Sécurité et fichiers	360
Permissions et droits d'accès	360
Arguments utilisateur	360
safe_mode et open_basedir.	360
La SPL et les fichiers	361
Obtenir des informations sur un fichier avec SplFileInfo.	361
Manipuler un fichier avec SplFileObject	362
Cas d'application	363
Outil de gestion documentaire simple	363

CHAPITRE 14

Gestion des flux	367
Exécution de programmes	367
Lancement sans interaction.	367
Lancement interactif.	371
Sécurité et programmes externes	377
Gestion des sockets réseau	377
Ouverture	378
Lecture et écriture.	379
Fermeture	379
Fonctions de contrôle	379
Gestion unifiée des flux	381
Types de flux gérés	382
Utilisation simple	386
Contextes	390
Filtres	392
Types personnalisés	394

Cas d'application	396
Système de paiement en ligne	396
Sauvegardes automatiques pour interface réseau	400
Conversion entre jeux de caractères	401
CHAPITRE 15	
Flux de sortie PHP	405
Principes et utilisations	405
Principe de fonctionnement	405
Exemples d'utilisation	406
Gestion du tampon de sortie	407
Début et arrêt de la mise en tampon	407
Récupération du contenu	407
Imbrication de tampons	409
Informations sur le tampon	410
Filtres automatiques	410
Compression des pages avec zlib	410
Conversion entre jeux de caractères	411
Filtres utilisateur	413
Automatisation	413
Tampon interne de PHP	414
Délai avant affichage	414
Vider le tampon	414
Autres tampons en jeu	415
CHAPITRE 16	
Envoyer et recevoir des courriels	417
De l'utilité de gérer des courriels	417
Webmail Open Source	418
RoundCube	418
IMP	419
Mise en œuvre	420
Prérequis techniques	420
Anatomie d'un courriel	421
Envoyer des courriels	422
Courrier électronique multimédia	425

Envoyer des courriels au format HTML	427
Envoyer des pièces jointes	429
Recevoir des courriels	434
Astuces et sécurité	439
Lancer un script à la réception d'un courriel	439
Vérification d'une adresse électronique	439
Espacer vos envois en masse	440
Bibliothèque Open Source : swiftmailer	440
Installation et initialisation	441
Écrire un courriel	441
Saisie des adresses électroniques	442
Interface fluide	442
Ajouter des fichiers joints	443
Utiliser du HTML dans votre courriel	443
Envoyer un courriel	444
Cas d'application	445
Gestion d'une lettre d'information	445

CHAPITRE 17

Travailler avec une base de données	447
Utilisation d'un SGBD	447
Qu'est-ce qu'un SGBD ?	447
Travailler avec un SGBD relationnel	448
Présentation de MySQL	449
Points forts/points faibles	449
Fonctionnalités	451
Types de tables MySQL	452
Outils d'administration Open Source	455
phpMyAdmin	456
Les commandes SQL	457
Créer une base de données	458
Créer des tables	459
Modifier des tables	465
Supprimer des tables	466
Insérer des données (INSERT)	466
Modifier des données (UPDATE)	468

Effacer des données (DELETE)	469
Remplacer des données (REPLACE)	470
Filtrer avec la clause WHERE	471
Sélectionner des données (SELECT)	472
Gérer les transactions	475
CHAPITRE 18	
Utiliser une base de données avec PHP	477
PDO, PHP Data Objects	477
Particularités	478
Utilisation des extensions natives à chaque SGBD	479
Utiliser votre base de données	481
Structure des classes de PDO	481
Prise en main rapide	482
Connexion au serveur de données	484
Structure du DSN	484
Utiliser des connexions persistantes	486
Gérer les erreurs de connexion	487
Fermer une connexion	487
Se connecter à plusieurs bases de données	488
Créer un fichier de configuration	489
Effectuer une requête	490
Requête de sélection	490
Requête d'insertion/modification	494
Requêtes invalides	495
Sécurité et échappements	496
Gestion des erreurs	498
Utiliser les exceptions	498
Gestion des transactions	499
Les requêtes préparées	500
Construction de la requête	502
Préparer une requête	503
Lier des données à des paramètres et exécution	503
Exploitation d'une requête de sélection	505
Fermeture de la requête préparée	505

Cas d'application	506
Gestion de publication	506

CHAPITRE 19

Erreurs et exceptions	515
Explications sur les erreurs	515
Qu'est-ce qu'une erreur ?	515
Que faire avec les erreurs ?	516
Les erreurs PHP	517
Description d'une erreur PHP	517
Les bases d'une gestion d'erreur	518
Niveaux d'erreurs et filtres	519
Créer une erreur manuellement	522
Journalisation des erreurs (log)	524
Personnaliser le gestionnaire d'erreurs	529
Les assertions	530
Description d'une assertion	530
Utilisation d'une assertion	530
Désactivation des assertions	531
Configuration des assertions	531
Personnalisation de la gestion	533
Les exceptions	533
Description d'une exception	533
Lancement d'une exception	535
Réception d'une exception	535
Filtrage des exceptions reçues	536
Propagation des exceptions	537
Utilisation des exceptions	539
Politiques de gestion d'erreur	542
Le développement	542
Être averti lors d'un problème	542
Toujours agir lors d'une erreur	542
Externaliser les alertes de sécurité	543
Gardez des traces sur le contexte	543

CHAPITRE 20

XML : concepts et SimpleXML	545
De l'utilité du XML	545
Gains apportés par XML	545
Exemples d'utilisation	546
Présentation et prérequis	546
Structure du XML	547
Principaux formats	550
Écrire un fichier XML manuellement	553
Écrire du XML avec XMLWriter	554
Prise en main rapide	555
Fonctionnalités avancées	556
Relecture et manipulations	557
Utilisation de SimpleXML	558
Import et export d'un document	558
Manipulation des éléments	559
Manipulation des attributs	563
Recherche Xpath	565
Extension des objets SimpleXML	566
Cas d'application	566
Lecture d'un fichier RSS	566

CHAPITRE 21

XML avancé	569
Relecture d'un XML avec SAX	569
Fonctionnement des événements	570
Initialisation	572
Réagir à des événements	573
Envoi des données et analyse	577
Manipulation avec DOM	579
Structure générale	580
L'objet document	581
Description d'un nœud	583
Navigation dans l'arbre	585
Gestion des attributs	589
Création de nœuds	592

Modification de l'arbre XML	593
Création d'un document complet	595
Recherche Xpath.	596
Initialisation du moteur.	596
Extension des classes DOM	598
Utilisation de Xinclude.	599
Validation et conformité	599
Transformation XML par XSLT	600
Utilisation du module XSL.	600
Initialisation	600
Chargement de la feuille de styles	600
Transformation	601
Paramètres de transformation	601
Extensions et interactions avec PHP.	602
 CHAPITRE 22	
Les services web	605
Introduction aux services web	605
Protocoles et technologies	605
Principe d'un appel à un service.	609
Utilisation simple (avec WSDL)	614
Créer un client SOAP	614
Créer un serveur SOAP.	616
Persistance	619
Cache WSDL	620
Utiliser SOAP sans WSDL	621
Créer un client SOAP sans WSDL.	621
Serveur SOAP sans WSDL.	623
Gestion des types et des structures	623
Compatibilité .Net et formats	625
Différents formats de message	625
Compatibilité avec un service .Net.	626
Autres détails et possibilités	627
Codage caractères	627
Définir des en-têtes SOAP	627
Utiliser un autre transport que HTTP	628

Gestion des erreurs	629
Erreurs reçues par un client SOAP	629
Utilisation des traces	629
Renvoyer une erreur dans un serveur	630
CHAPITRE 23	
Les templates	631
De l'utilité des templates	631
Moteurs de templates Open Source	632
Une solution légère : PHPLib	632
Le couteau suisse : smarty	632
Un système original : Templeet	633
Différentes approches	633
L'approche PHP natif	634
L'approche search&replace	636
L'approche par composants	638
Utilisation de XML et XSLT	640
Analyse et choix	641
Pérennité de la solution retenue	641
Simplicité pour les graphistes	642
Simplicité pour les développeurs	642
Les performances du moteur	643
Bibliothèques Open Source	643
PHPLib	644
Smarty	646
Templeet	652
CHAPITRE 24	
Les systèmes de cache	657
De l'utilité des caches	657
Outils de cache Open Source	658
Mise en œuvre	658
Les caches globaux	658
Cache d'une page HTML	659
Cache de fichiers de différents types	661
Cache de configuration	662

Cache des données utilisateur	662
Cache par session	663
Les caches HTTP	663
Dates de mises à jour des fichiers	664
Utilisation des serveurs proxys	665
Utiliser la date d'expiration	667
Mise à jour du cache	667
Détection de la modification	668
Temps de validité	668
Sites semi-statiques	669
Pear::Cache	669
La classe générique	670
Classe pour le Cache HTML	672
Autres caches	673
Pear::Cache_Lite	674
Utilisation	674
Spécialisations	675
Étude de cas	676
Cache pour un site d'actualité	676

CHAPITRE 25

Gestion des images	679
Utilité de la gestion d'images	679
Prérequis techniques	680
Initialisation et utilisation	680
La création du modèle de l'image	681
Libérer les ressources mémoire	683
Affichage de l'image sur le navigateur	683
Enregistrer l'image dans un fichier	685
Travail sur une image	685
Le référentiel	686
Tracer des formes	686
Écrire du texte	689
Copie d'une zone d'image	692
Gestion de la palette de couleurs	693
Connaître la taille d'une image	694

Astuces et remarques	695
Éviter les fausses couleurs	695
Limite de temps	695
Malvoyants et référencement	695
La bibliothèque Open Source JpGraph	696
Installation et configuration	697
Architecture de JpGraph	698
Création d'un graphique	698
Envoi et enregistrement de l'image	699
Gérer les polices de caractères	699
Propriétés et méthodes communes	700
Les graphiques à base de lignes	701
Les graphiques en camemberts	703
D'autres types de graphiques	706
Étude de cas	707
Redimensionner des images	707
Superposer des images	710
CHAPITRE 26	
Expressions rationnelles	713
Syntaxe	713
Protections et échappements	714
Délimitation et présentation	714
Chaîne de recherche simple	715
Construction d'expression	716
Gestion des occurrences multiples	719
Assertions	720
Captures	723
Modificateurs	724
Les fonctions	725
Chercher une correspondance	725
Faire des remplacements	726
Échappement et protections	729
Performances	730
Fonctionnement du moteur	730
Stratégies	731
Boucles infinies	731

CHAPITRE 27

Sécurité	733
Qu'est-ce que la sécurité ?	733
Préoccupations du gestionnaire	734
Préoccupations de l'utilisateur	734
Pourquoi parler de l'utilisateur ?	735
Configuration et sécurité	736
Interface avec le serveur web	736
Safe_mode et restrictions	738
Échappement automatique	740
Variables globales	741
Sessions et identifiants	741
Mises à jour du logiciel	743
Stockage des données et fichiers	743
Sécurité de l'application	745
Vérification des entrées utilisateur	745
Éviter les principales attaques	749
Emplacement des contrôles	753
Gérer les erreurs	756
Sécuriser les sessions	756
Chiffrement et sécurité	757
Bonnes habitudes	760
Vérifiez vos résultats	760
Ne croyez pas l'utilisateur	761
N'exagérez pas	762
Faites faire un audit externe	763

CHAPITRE 28

Outils de développement PHP	765
Éditeurs de texte & IDE	765
UltraEdit	766
PHPEdit	768
Eclipse	774
Le Zend Studio	776
Un outil de modélisation/RAD : UML2PHP5	779

CHAPITRE 29

Les frameworks	783
Ce qu'est un framework	783
Un cadre de travail	783
La séparation du code en couches	784
Les avantages d'un framework	784
Quelques frameworks disponibles en Open Source	785
Symfony	785
Zend Framework	786
Les autres	788
Courte introduction à Symfony	788
Installation	788
Configuration de l'application	790
Étude de l'architecture de l'application	791
Création et activation d'un nouveau bundle	792
Mise en place de la couche de persistance avec Doctrine 2	794
Découvrir l'architecture MVC de Symfony	798
Créer la liste des événements	800
Consulter le détail d'un événement	802
Créer des URL et des liens hypertextes	805
Conclusion	806

CHAPITRE 30

Migrations	807
Importance de la migration	807
Depuis PHP 4	807
Depuis PHP 5	808
Quand migrer ?	808
Le processus général	810
Lister les dépendances tierces	810
Lister les incompatibilités	810
Changements majeurs	811
Changements mineurs	811
Tester le résultat	812
Faire un test de performance	812

Les changements majeurs	813
Données utilisateur et système	813
Passage de variables par référence	816
Copies et références vers des objets	817
Utilisation de MySQL	819
Utilisation de XML	820
Les extensions	821
Configuration	821
Les changements mineurs	822
Si vous migrez depuis PHP 4	823
PHP 4 et PHP 5	826
Changements après PHP 5.0.0	828
En prévision de PHP 5.4	832
Et PHP 6 ?	833
 CHAPITRE 31	
PHAR (PHP Archive)	835
Présentation générale	835
Formats et fonctionnalités	835
Installation et configuration	836
Utilisation	836
Création d'une archive PHAR	837
Initialisation	838
Ajouter un fichier de démarrage (archives exécutables)	839
Supprimer le fichier de démarrage	841
Ajouter ou remplacer des fichiers	842
Supprimer des fichiers	843
Lister les fichiers présents	844
Signer une archive	844
Métadonnées de l'archive	845
Ajouter ou modifier des métadonnées	845
Lire les métadonnées présentes	846
Supprimer des métadonnées existantes	846
Compression des archives	847
Compresser l'archive complète	847
Compresser les fichiers individuellement	847
Décompresser une archive	848

Accéder à un fichier externe depuis une archive	850
Extraire les fichiers contenus dans une archive	851
Trucs et astuces	852
Obtenir le chemin ou l'URL d'accès à une archive PHAR	852
Constantes	852
Alias	853
Interface \countable	853
Index des fonctions PHP	855
Index	857

Avant-propos

Pourquoi ce livre ?

Pourquoi écrire et faire vivre un livre si son sujet n'est pas une affaire de passion ? En effet, pour nous, le Web et PHP sont des affaires de cœur. Nous allons vous transmettre non seulement un savoir, mais aussi une expérience et une passion.

PHP peut être considéré comme l'un des fers de lance du monde Open Source. Toute l'image de cette philosophie de partage et d'entraide s'exprime à travers lui. Et si à une belle idée on associe un produit fiable, stable, complet et étendu, pourquoi hésiter ?

En dépit de ses atouts, PHP a été longtemps perçu par les professionnels comme un outil pour pages personnelles ou petits projets. Certes, il est adapté à ce type de missions, mais son spectre d'action est nettement plus vaste. Heureusement, grâce à ses qualités intrinsèques et à sa communauté qui a réussi à se faire entendre et à séduire, les mentalités ont fini par évoluer et PHP a été élevé à sa juste valeur.

Ce livre, nous l'avons pensé, écrit, et nous le faisons évoluer pour les développeurs pointilleux désirant exploiter au mieux les capacités de PHP. Sans le rendre inaccessible aux débutants, nous souhaitons qu'il soit utile à des développeurs professionnels ou d'un niveau avancé. Nous avons tous deux des profils différents, l'un très technique et puriste, l'autre orienté vers le fonctionnel, la vulgarisation et la pédagogie. Le résultat se veut donc très pointu et très vaste, tout en adoptant une approche pédagogique.

Ces pages ont été conçues de façon à souligner les nouveaux ajouts et à fournir une référence utile au jour le jour pour les développeurs PHP. Contrairement à d'autres ouvrages qui se fondent massivement sur l'excellente documentation de PHP (visible en ligne et à jour sur <http://fr.php.net>), nous avons souhaité réaliser un livre qui apporte une réelle valeur ajoutée, dépassant le simple étalage des fonctions et des paramètres.

Structure de l'ouvrage

Cet ouvrage s'articule autour des thèmes abordés lors du développement d'une application web. Chaque chapitre est centré sur un de ces thèmes. Il décrit les différentes fonctionnalités PHP qui s'y rapportent, mais aussi tout ce qui les entoure et qui permettra de

les mettre en œuvre. Des exemples concrets, des cas d'applications pratiques et des retours d'expériences seront régulièrement présentés.

La première partie du livre fait office d'entrée en matière.

- Le **chapitre 1** donne toutes les informations sur la plate-forme PHP, sa diffusion et les ressources d'aide que vous pourrez trouver, francophones et internationales.
- Le **chapitre 2** détaille les options de configuration les plus importantes et les procédures d'installation, sous Unix et Microsoft Windows.

La partie suivante concerne les fonctionnalités de base du langage. On y trouve les rappels sur la syntaxe et les structures, puis l'interface avec les pages web via les formulaires ou les cookies. Cette partie enseigne aux débutants les bonnes bases de PHP. Les développeurs confirmés pourront, eux, y trouver une référence avec quelques astuces et détails utiles.

- Le **chapitre 3** fait un rappel sur les structures de base du langage PHP : types de données, affectation, organisation du code, etc.
- Le **chapitre 4** montre les traitements de base permis par PHP : les différentes boucles et conditions.
- Le **chapitre 5** détaille les différentes fonctions de gestion des chaînes de caractères.
- Le **chapitre 6** se focalise sur la gestion des tableaux et les fonctions afférentes.
- Le **chapitre 7** présente les quelques fonctions usuelles qui ne se rapportent pas à un sujet particulier et qui sont souvent utiles lors de développements.
- Le **chapitre 8** décrit l'interaction entre PHP et les formulaires HTML (variables, fichiers), ainsi que les superglobales PHP permettant leur manipulation.
- Le **chapitre 9**, dernier de cette première partie très orientée vers la référence, complète le précédent en s'intéressant à l'environnement autour de PHP : principalement la communication avec le serveur web, le système et le réseau.

La troisième partie entre dans le cœur du sujet en se focalisant sur différents thèmes rencontrés dans le cadre du développement d'applications avancées. Le développeur confirmé y trouvera matière à progresser.

- Le **chapitre 10** offre une description avancée des cookies, de leur utilisation et de leur environnement. On y retrouvera aussi quelques informations liées à la sécurité.
- Le **chapitre 11** évoque les sessions. Outre la description simple de leur utilisation, nous abordons une réflexion globale sur les sessions, leur utilité et leur sécurité. Les développeurs confirmés y trouveront les informations pour mettre en œuvre leur propre gestionnaire de sessions.
- Le **chapitre 12** présente les fonctionnalités de PHP relatives à la programmation orientée objet. Une description complète des fonctionnalités y est proposée, y compris celles ajoutées par PHP 5.3.

- Le **chapitre 13** décrit en détail la gestion des fichiers : lecture, écriture, manipulations, fichiers distants, etc.
- Le **chapitre 14** étend les notions abordées avec les fichiers pour manipuler tout type de flux de données : sockets réseau, exécution de programmes externes et flux personnalisés.
- Le **chapitre 15** s'intéresse à la gestion du tampon de sortie de PHP : appliquer un filtre sur les données envoyées au navigateur, manipuler le flux de sortie pour compresser les pages web, etc.
- Le **chapitre 16** détaille tout ce que vous devez savoir concernant l'envoi et la réception de courriels : de l'utilisation pour envoyer un simple message texte jusqu'à la description des courriels HTML ou avec pièces jointes.
- Le **chapitre 17** est dédié au langage SQL et aux SGBD en général, avec une approche poussée du cas de MySQL.
- Le **chapitre 18** présente en détail comment communiquer entre PHP et une base de données en utilisant PDO (*PHP Data Object*).
- Le **chapitre 19** est dédié à la gestion des erreurs avec PHP. La première partie décrit la gestion des erreurs classiques telles qu'on peut les voir dans PHP 4 et des assertions. La seconde partie décrit une nouveauté de PHP 5 : les exceptions. D'autres points comme la configuration, les journaux d'erreur ou la politique de gestion des erreurs sont aussi abordés.
- Le **chapitre 20** présente une autre nouveauté de PHP 5 : la gestion XML avec SimpleXML. Les notions basiques y sont abordées, ainsi que tout ce dont vous avez besoin pour lire et manipuler rapidement du XML.
- Le **chapitre 21** complète le précédent en donnant les méthodes pour les manipulations avancées que vous pourriez avoir à faire avec XML : SAX, DOM, XSLT, etc.
- Le **chapitre 22** traite des services web et particulièrement de SOAP.
- Le **chapitre 23** traite de la dissociation de la logique métier et du visuel : les gabarits, ou *templates*.
- Le **chapitre 24** aborde toutes les problématiques de la gestion des caches. Il vous donne toutes les clés pour trouver ou créer le système adapté à vos besoins.
- Le **chapitre 25** détaille l'utilisation de l'extension GD. Elle vous permettra de produire ou de manipuler facilement des images, des photos diagrammes ou des graphiques avec PHP.
- Le **chapitre 26** se concentre sur l'utilisation des expressions régulières. La syntaxe et l'utilisation des expressions compatibles Perl reconnues par PHP y sont décrites en détail.

La quatrième et dernière partie traite des sujets annexes lors de vos développements : la sécurité et les outils.

- Le **chapitre 27** fait un tour des aspects de la sécurité à prendre en compte lors du développement d'une application. Vous y trouverez des exemples de failles ou de problèmes fréquents, ainsi que les bonnes habitudes pour les éviter.
- Les **chapitres 28** et **29** ajoutent une description des différents outils de développement pour PHP, ainsi qu'une revue rapide des *frameworks* intéressants.
- Le **chapitre 30** est dédié aux migrations, de la version 4 à la version 5, mais aussi entre les différentes versions de PHP 5.
- Enfin, le **chapitre 31** clôture ce livre en présentant les archives PHAR, leur intérêt et la façon de les créer, les manipuler et les utiliser.

Remerciements

Nous tenons à remercier tous ceux qui nous ont aidés à rédiger ce livre :

- les familles, proches et amis, pour leur soutien et leur patience pendant ces longs mois de rédaction et de réflexion ;
- les éditions Eyrolles, pour avoir cru en notre projet et l'avoir soutenu dès le départ ;
- Sarah Gedon, Romain Bourdon, Sarah Haim, Grégoire Cachet, Valérie Poinssotte et Stéphane Deschamps pour leurs multiples aides dans le développement des divers chapitres ;
- Hugo Hamon et Frédéric Hardy qui nous ont grandement aidés dans les refontes successives du livre ;
- dans le désordre, Christophe Gesché (Moosh), Jérôme Renard, François Dume, Vincent Brouté, Alain Gazalet, Eudes Robichon, Frédéric Bordage, Guillaume Bouchard, Julien Jackubowski, Yoan Blanc, Laurent Jouanneau, Damien et Ghislain Seguy, Quentin Sinagra, Rémi Pauchet, KDO, Xavier Langlet, Jean-Eudes Amrein, Raphaël Rousseau et Stéphane Raviart pour les diverses relectures qu'ils ont pu faire ;
- tous les lecteurs des précédentes éditions qui, par leurs retours, nous ont permis d'améliorer cet ouvrage ;
- ... et tous les autres dont nous n'avons pas le nom complet ou les coordonnées, ou que nous avons simplement oubliés dans la précipitation qui a précédé l'impression de cette page.

Merci à tous, car sans vous, ce livre n'aurait peut-être pas vu le jour.

Éric Daspét et Cyril Pierre de Geyer

1

Qu'est-ce que PHP ?

PHP (*PHP Hypertext PreProcessor*) est à la fois un langage de programmation et une plate-forme d'entreprise sur laquelle on peut poser des briques logicielles (CRM, GED, CMS...).

Son positionnement se situe au niveau du développement d'applications web dynamiques allant du simple site à un outil de l'entreprise, voire un portail à très gros trafic. De façon générale, PHP va fournir du contenu HTML dynamique. On peut par exemple lui faire créer le contenu de pages HTML suivant différents paramètres : l'âge d'un visiteur, sa catégorie socioprofessionnelle, ses droits sur l'application, des mots-clés qu'il aura indiqués dans un moteur de recherche, des actualités du jour, etc.

Les capacités de PHP ne s'arrêtent pas à la création de pages web. Il est aussi possible de manipuler des images, de créer des fichiers PDF, de communiquer via des services web, de se connecter à des bases de données ou des serveurs LDAP, et même d'instancier des objets Java. Un module annexe lui permet également de fournir des interfaces graphiques classiques (client lourd, sans navigateur ou serveur web), via GTK.

Les fonctionnalités de PHP permettant de sortir de l'ordinaire des sites web sont très nombreuses. Dans ce chapitre, nous vous présenterons ses possibilités, ses caractéristiques et son historique. Enfin, nous aborderons PHP du côté français, c'est-à-dire en mettant en avant les ressources mises à disposition par la communauté francophone.

Introduction à PHP

Un langage Open Source

PHP est à l'origine un langage de script conçu spécifiquement pour agir sur les serveurs web. En ajoutant quelques lignes de PHP à une page HTML, le serveur exécute les

instructions correspondantes pour écrire du code HTML à la place. Le résultat (le code HTML initial ajouté à celui produit par PHP) est envoyé au navigateur. Cela permet par exemple d'afficher la date du jour à un endroit bien précis du visuel. On parle alors de page dynamique.

Dans l'exemple suivant, PHP ajoute une chaîne de caractères au milieu du code HTML :

```
<html>
  <head>
    <title>Exemple</title>
  </head>
<body>
  <p>
    <?php
      echo "Ceci est une syntaxe PHP";
    ?>
  </p>
</body>
</html>
```

PHP dispose de près de 4 000 fonctions utilisables dans des applications très variées et couvre pratiquement tous les domaines en rapport avec les applications web. Par exemple, presque tous les SGBD du marché (Systèmes de Gestion de Bases de Données) peuvent s'interfacer avec PHP, qu'ils soient commerciaux ou qu'ils viennent du monde du logiciel libre.

Licence et téléchargement

PHP est distribué sous une licence propre qui permet sa rediffusion, son utilisation et sa modification librement et gratuitement. Il peut être téléchargé depuis le site web officiel sur <http://www.php.net/> ou un de ses miroirs tel que <http://fr.php.net/>.

Exécution

L'exécution de PHP est similaire à celle de Java ou des langages .NET, c'est-à-dire que les scripts sont convertis en un langage intermédiaire (*byte code*) avant d'être exécutés. Toutefois, à la différence de ces langages, le code intermédiaire de PHP est recréé à chaque exécution et ne peut pas être diffusé. Du point de vue utilisateur, on exploite directement le code source : il n'y a pas d'étape de compilation.

Courbe d'apprentissage

Reprenant une syntaxe claire et familière puisque très proche de celle du langage C, PHP est un langage dont la prise en main est généralement très rapide. Il est facile d'en apprendre les bases mais il est difficile de le maîtriser pleinement. Effectivement, connaître et utiliser toutes les fonctionnalités et concepts de PHP nécessite un apprentissage poussé.

Que faire avec PHP ?

La principale utilisation que l'on peut avoir de PHP est l'utilisation d'un langage de script traité côté serveur pour la création de pages web. Cette utilisation sur serveur web est la principale mais PHP peut également être utilisé pour d'autres types de développements.

Fonctionnement couplé à un serveur web

Le fonctionnement sur un serveur web est l'application la plus répandue. Trois composants entrent en jeu : un serveur web (le plus souvent Apache ou IIS), le module PHP et un navigateur web. Lorsque le serveur web reçoit une demande de page, PHP en élabore le contenu avant de l'envoyer au navigateur. Ce mode de fonctionnement permet de créer des sites Internet dynamiques ou de s'interfacer avec des progiciels pour gérer la logique métier de l'entreprise.

Applications en ligne de commande

Vous pouvez utiliser PHP de façon autonome, sans serveur web, en ligne de commande. Pour cela, il vous suffit de faire appel à l'exécutable `php`. Cela peut parfois être utile pour réaliser des actions simples sur votre ordinateur (par exemple, changer automatiquement le nom de plusieurs centaines de fichiers) sans nécessiter la présence de tout un contexte web. Pour automatiser des actions récurrentes, vous pouvez coupler son utilisation au gestionnaire des tâches (serveur `cron` sous Linux). Il suffit dans ce cas de passer à l'exécutable PHP le fichier contenant le script à exécuter : `php -f rename.php`.

Services web

PHP permet de créer et de consommer des services web. Ce type d'application met votre contenu à disposition d'autres personnes. Ainsi, tels Amazon, Google ou Yahoo!, vous pourrez créer et distribuer vos propres applications que d'autres utiliseront. On parle alors d'applications en « marque blanche ». Amazon, par exemple, vous permet de reprendre son catalogue, de le mettre à vos couleurs et de vendre ses produits comme s'il s'agissait des vôtres. PHP vous permet autant de gérer et de produire des services web que d'en utiliser.

Applications graphiques

PHP dispose d'une extension permettant de produire des applications graphiques traditionnelles. Il n'y a alors ni serveur web ni navigateur, et l'application s'exécute entièrement sur le poste client. L'extension nécessaire n'est pas incluse par défaut, mais vous pouvez la récupérer sur un site dédié : <http://gtk.php.net/>. Les interfaces avec le SGBDR fichier SQLite permettent alors de réaliser des applications sans serveur ni service tiers.

Particularités de PHP

Les principaux « concurrents » de PHP sont .NET , Perl, Python, Ruby et Java.

Globalement, il faut garder en tête qu'à chaque problème correspond sa solution et qu'il est difficile de dire que tel langage ou tel autre est meilleur de façon générale. Cependant, PHP 5 dispose par rapport à ses concurrents de quelques particularités et avantages significatifs.

De nombreux connecteurs techniques

PHP intègre des possibilités de connexion à la majorité des bases de données (Oracle, SQL Server, MySQL, dBase, ODBC, etc.), annuaires (LDAP, etc.) et systèmes de paiement en ligne (VeriSign, Cybercash, Crédit Mutuel, etc.).

C'est particulièrement intéressant quand on sait que près de 40 % de la charge de développement d'une application est liée à l'intégration d'applications ou de sources de données existantes (selon IDC, cabinet de conseil et d'études sur les marchés des nouvelles technologies de l'information).

L'essentiel des protocoles et des formats qu'on peut rencontrer sur Internet ou intranet sont aussi pris en charge : TCP, HTTP, SMTP, LDAP, IMAP, POP, SSL, Soap, XSLT, XML, PDF, etc.

Peu de connecteurs applicatifs

Bien que pouvant s'interfacer avec SAP, Lotus Notes, IBM iseries et d'autres progiciels, PHP ne dispose pas d'un grand nombre de connecteurs applicatifs. On peut regretter par exemple l'absence de connecteurs vers les principaux MOM du marché (*Message Oriented Middleware*) tels que Tibco, MQseries ou Microsoft MSMQ. On trouve toutefois un connecteur pour SAP qui permet d'exécuter les différentes fonctions du progiciel.

La possibilité pour PHP de se connecter directement au *backend* (interfaces internes des logiciels) et aux bases de données permet de compenser en partie ce manque.

Les performances de PHP

PHP est extrêmement performant et fiable, même selon les critères d'application critiques. Avec un seul serveur standard, on peut répondre à des millions de requêtes par jour. Pour des sites à très fort trafic, il existe diverses solutions permettant d'optimiser et d'améliorer les performances globales de PHP.

Des sites ou des applications critiques utilisent PHP (*Le Monde, Le Figaro, TV5, Yahoo, TF1, Canal +...*). Il s'agit maintenant d'une solution reconnue comme viable autant du côté stabilité et fiabilité que du côté des performances.

Les bases de données reconnues par PHP

PHP 5 contient des connexions natives vers la plupart des Systèmes de Gestion de Bases de Données (SGBD). Depuis la version 5, PHP dispose également d'une base de données

directement intégrée : SQLite. Voici une liste non exhaustive des bases de données reconnues par PHP : Microsoft SQL server, Oracle, PostgreSQL, MySQL, Sybase, SQLite, FilePro, Informix, Interbase, mSQL, dBase, Empress, et bien d'autres.

De plus, le standard ODBC (*Open DataBase Connectivity*) et les fonctions ODBC de PHP permettent de se connecter à n'importe quelle base de données possédant un pilote ODBC.

Services web et interopérabilité

PHP est le champion de l'intégration bas niveau. Il est capable d'instancier des objets COM, des classes Java, Python ou .NET. L'intégration de bibliothèques C via des modules PHP est elle aussi aisée.

PHP dispose également nativement d'une couche SOAP et d'une couche XML-RPC. Elles permettent de créer ou de consommer des services web très simplement. Vous pouvez par exemple vous connecter au moteur de recherche Google ou au système d'Amazon pour y effectuer des recherches.

Les flux XML associés aux parseurs XSL/XSLT vous permettent de travailler avec d'autres systèmes d'information. Des connectivités SNMP, LDAP sont aussi disponibles. Les différents modules de PHP couvrent une base extrêmement large sur tout ce qui peut être en interaction avec un script web. Il serait surprenant que vous n'y trouviez pas de quoi répondre à vos besoins.

Bibliothèques intégrées

PHP a été conçu pour le Web et, par conséquent, il dispose de nombreuses fonctions permettant d'effectuer la majorité des actions s'y rapportant.

Il est par exemple possible de créer des fichiers PDF, produire des images à la volée, se connecter et communiquer avec d'autres serveurs web ou FTP, envoyer et recevoir du courrier électronique. Toutes ses bibliothèques bénéficient de fonctions de haut niveau permettant au programmeur de se concentrer sur son application au lieu de gérer les détails de chaque composant.

La portabilité

PHP est disponible pour plusieurs systèmes d'exploitation. Il fonctionne sous MS Windows (toutes versions supérieures à Windows XP depuis PHP 5.3) et l'essentiel des versions d'Unix ou associés (par exemple Solaris, Linux, OpenBSD, FreeBSD, Mac OS X, etc.). Votre code pourra être utilisé sur toutes ces plates-formes de la même façon et quasi sans modification.

Coûts de licence

PHP est gratuit. Vous pouvez, à tout moment, vous procurer la dernière version sur le site : <http://www.php.net>, sans payer quoi que ce soit. Cependant le prix du logiciel PHP n'est pas le seul à entrer en compte. Il faut aussi prévoir le prix du système d'exploitation,

d'une éventuelle base de données, du serveur web, etc. L'avantage de PHP est qu'il peut, comme indiqué précédemment, être utilisé dans la majorité des cas. Ainsi, vous pourriez autant l'utiliser avec une plate-forme sous Linux qu'avec une plate-forme sous Windows, voire sur AS400. Dans cette optique, vous pouvez utiliser PHP couplé à un serveur Linux et une base de données MySQL sans déboursier un centime d'euro.

Coûts de développement

Un développement fait en PHP est généralement plus rapide qu'un développement effectué sous J2EE ou .NET, le code étant plus court et moins complexe. De plus, actuellement, le coût journalier d'un bon développeur PHP est moins élevé que celui d'un bon développeur Java.

Ainsi, globalement, les coûts de développement PHP sont généralement moins importants que les coûts induits par l'utilisation des alternatives.

Le code source

Le code source de PHP est disponible gratuitement. À l'inverse des produits commerciaux dont les sources ne sont pas distribuées, vous avez la possibilité de modifier tout ou partie des sources pour adapter PHP à vos besoins spécifiques. Le produit modifié peut être vendu et redistribué librement suivant vos propres conditions.

Dynamisme de la communauté PHP

La communauté PHP est estimée par la société Zend à près de 4 500 000 développeurs courant 2007. Elle est très organisée et très réactive. L'annonce d'une faille de sécurité implique généralement un correctif dans la journée. De plus, de nombreuses personnes développent des outils Open Source de très bonne facture et les proposent au public.

Historique

Contrairement à d'autres langages comme le C, le C++, voire le Perl, PHP est un langage assez jeune. Son évolution sur quelques années en a fait l'un des langages les plus importants du Web.

PHP/FI

PHP/FI a été créé en 1995 par Rasmus Lerdorf. À l'origine, il s'agissait d'une bibliothèque de scripts fonctionnant sous Perl, dont l'objectif était, entre autres, de permettre à son auteur de savoir qui venait consulter son CV sur son site personnel. Rasmus donna donc à cette bibliothèque son premier nom : *Personal Home Page Tools*.

Petit à petit, la bibliothèque Perl s'est muée en une implémentation directement en C, l'objectif étant des gains de performances et des possibilités plus poussées : communiquer avec les bases de données, créer des applications dynamiques pour le Web, etc.

À ce stade, Rasmus décida de proposer son code à la communauté afin que tout le monde puisse l'utiliser et en profiter, voire contribuer à son développement.

PHP/FI signifiait à cette époque *Personal Home Page / Forms Interpreter* pour indiquer, chose rare à l'époque, que PHP/FI gérait les formulaires (FI pour Interpréteur de formulaire). Ses principales caractéristiques étaient la simplicité d'insertion dans du HTML, une syntaxe proche du Perl et un système d'interprétation des variables de formulaires.

Bien que très jeune, le langage disposait déjà de nombreux adeptes. En 1997, on estimait l'audience à plusieurs milliers d'utilisateurs. Près de 50 000 domaines avaient installé PHP (soit 1 % des noms de domaines).

PHP/FI 2.0 fut publié officiellement en novembre 1997, après avoir passé l'essentiel de sa vie en version bêta. Peu de temps après, une version alpha de PHP 3.0 était publiée.

PHP 3

PHP 3.0 n'est pas réellement une suite à PHP/FI mais plutôt une refonte. En 1997, Andi Gutschman et Zeev Suraski (fondateurs de Zend : combinaison des prénoms Zeev et Andi) essayèrent d'utiliser PHP/FI dans le cadre du développement d'une application de e-commerce, mais les performances n'étaient pas suffisantes. Ils décidèrent de réécrire PHP/FI de façon complète.

PHP 3.0 a été la première version de PHP assez fonctionnelle et stable pour être mise en production sur de véritables projets. Afin d'assurer une continuité avec PHP/FI, Rasmus rejoignit le projet PHP 3.0, qui devint le successeur officiel de PHP/FI 2.0.

Parmi les nouvelles fonctionnalités de PHP 3.0, la possibilité d'y intégrer des extensions fut sûrement celle qui lui permit de connaître un tel succès. En effet, une API modulaire donna la possibilité à n'importe quel développeur de créer ses propres modules et de les partager avec l'ensemble de la communauté. Des modules permettant de créer des images dynamiquement ou de travailler sur des fichiers PDF sont ainsi apparus.

Avec cette nouvelle mouture, PHP devenait un langage de programmation à part entière et se devait de prendre un nom plus professionnel. C'est ainsi que PHP devint *PHP Hypertext Preprocessor*.

Au bout d'une dizaine de mois de test et de débogage, la première version officielle de PHP 3.0 fut lancée en juin 1998. À la fin de cette même année, PHP était déjà utilisé sur des centaines de milliers de sites. On estime que PHP 3.0, à son apogée, était installé sur 10 % du parc mondial des serveurs web.

PHP 4

Juste après la publication de PHP 3.0, Andi et Zeev se remirent au travail pour réécrire totalement le moteur de PHP car, malgré ses fonctionnalités et sa stabilité, ils n'étaient pas satisfaits de ses performances.

Ils commencèrent donc à travailler sur ce qu'on appellera par la suite le *Zend Engine*. Une première version de ce moteur fut publiée dans le courant de l'année 1999, mais ce n'est qu'en mai 2000 qu'il fut officiellement intégré à PHP dans sa nouvelle version : PHP 4.0.

En plus de ce nouveau moteur apportant des performances beaucoup plus élevées, les principales évolutions de PHP 4.0 par rapport à son prédécesseur tenaient à sa prise en charge des sessions HTTP et de nombreux serveurs web, ainsi qu'à la mise en tampon des sorties et à une sécurité accrue des informations visiteurs.

PHP 5

La première version stable de PHP 5 a fait son apparition en 2004. L'objectif était double : rendre PHP plus professionnel, mais également le simplifier. Les versions 5.1 et 5.2, quant à elles, sont respectivement sorties en 2005 et 2006. Par rapport à la version 4, les principales nouveautés sont :

- l'intégration du Zend Engine 2, qui amène une prise en charge complète de la programmation orientée objet ;
- la refonte de la prise en charge de XML ;
- l'intégration de la base de données SQLite ;
- la simplification des principales tâches courantes ;
- l'apparition d'un socle commun pour la gestion des appels aux bases de données : PHP Data Object (PDO) ;
- l'implémentation de l'extension Filter qui permet aux développeurs de valider et de filtrer plus efficacement des données issues de sources non sécurisées comme les saisies de formulaires.

PHP 5.3

Dans la foulée de la sortie de PHP 5, le développement de PHP 6 a débuté avec pour objectif principal de prendre en charge Unicode, une norme informatique qui attribue un identifiant unique à chaque caractère de chaque système d'écriture existant dans le monde, afin de faciliter son affichage et sa manipulation par les ordinateurs. La solution technique retenue pour son implémentation a cependant été rejetée par les développeurs de PHP, qui ont préféré se concentrer sur le développement de la branche 5. En conséquence, ils ont introduit dans PHP 5.3 la plupart des fonctionnalités qui devaient être intégrées dans PHP 6, hormis la reconnaissance de la norme Unicode.

La première version stable de PHP 5.3 est sortie en 2009 et a apporté notamment la gestion des espaces de noms, des fonctions anonymes, ainsi que la résolution statique à la volée (*Late Static Binding* ou LSB).

Le développement de PHP 6, tel qu'il était prévu initialement, a été quant à lui abandonné en mars 2010 et, à cette occasion, il a été décidé que les prochaines versions du langage seront basées sur la branche 5, plus mature et plébiscitée par les développeurs.

Mode de développement du projet PHP

Le mode de développement de PHP, fondé sur le travail collaboratif, impressionne. Il n'est pas rare que les gens s'étonnent qu'un tel outil ait pu être développé bénévolement.

C'est pourtant le cas ; cependant, pour qu'un tel système fonctionne, une hiérarchie se doit d'être définie et suivie tout en restant souple.

Les différentes équipes

Plusieurs équipes travaillent au développement de PHP :

- équipe de développement (500 personnes) ;
- équipe qualité (250 personnes) ;
- équipe de documentation (120 personnes) ;
- équipe de traduction (120 personnes).

Étant donné que de nombreux contributeurs participent à plusieurs équipes, on estime leur nombre total à 700 personnes. Une illustration de l'organisation est donnée à la figure 1-1.



Figure 1-1

Déroulement du développement

Note

On notera cependant que ces contributeurs ne travaillent pas en permanence ni toujours ensemble, mais à leur rythme et en alternance. Ainsi, on peut estimer qu'environ 10 % des inscrits travaillent à un moment donné.

L'équipe de développement

Les sorties (*releases*) sont généralement gérées par un RM (*Release Master*) qui joue le rôle de l'organisateur. Il est éventuellement aidé par un RMB (*Release Master Bitche*), dont le rôle est de gérer les tâches ingrates : servir d'avocat du diable, recueillir les critiques et les bogues, etc.

La désignation d'un RM se fait sur une base de volontariat et par approbation de ses pairs.

Les développeurs ont longtemps utilisé l'outil CVS pour gérer les différentes versions. Depuis 2008, ils utilisent le logiciel Subversion.

Note

Les contributeurs ne travaillent pas en permanence ni toujours ensemble, mais à leur rythme et en alternance. Ainsi, on peut estimer qu'environ 10 % des inscrits travaillent à un moment donné

L'équipe de gestion qualité

Une fois une version candidate à la mise en ligne prête, l'équipe de qualité entre en jeu. Son travail consiste à effectuer des batteries de tests sur l'ensemble de la version candidate. Une version candidate n'est jamais proposée sans qu'elle ait passé l'ensemble des tests.

L'équipe de documentation

L'équipe de documentation travaille à la mise en place de documentation pour les utilisateurs. La première version étalon se fait en anglais.

L'équipe de traduction

Pour que chacun puisse accéder facilement à l'information dans sa propre langue, des équipes internationales œuvrent à traduire la documentation dans leur langue maternelle. On remarquera d'ailleurs que le site <http://php.net> met automatiquement à disposition la documentation dans votre langue.

Principaux apports de PHP 5

Les principaux apports des versions 5 à 5.2 de PHP

La programmation orientée objet

PHP 5 a fait son apparition en 2004. Sa principale nouveauté résidait dans la nouvelle mouture de son moteur : le *Zend Engine 2*. Ce nouveau moteur, encore utilisé aujourd'hui, permet de gérer dans leur ensemble les aspects de la programmation objet, remédiant ainsi à ce que certains considéraient comme un défaut de PHP 4.

Refonte et simplification de XML

Les autres nouveautés concernaient la gestion de XML. La version 4 de PHP impliquait une utilisation relativement lourde pour qui souhaitait manipuler des flux XML. Avec la version 5, deux nouveautés ont révolutionné sa manipulation :

- l'intégration d'un nouveau gestionnaire XML, basé sur la bibliothèque libxml2, qui amène une implémentation DOM standard complète ;
- l'extension SimpleXML.

La première permet à PHP de traiter tous les aspects de la manipulation XML, avec la complexité que cela implique.

La seconde s'adresse à tous les traitements XML simples. Il n'est plus obligatoire de passer des opérations compliquées pour récupérer les données de fichiers XML.

Intégration de la base SQLite

Les développeurs de PHP 5 ont également intégré au langage un moteur de bases de données, nommé SQLite. Ce dernier dispose de nombreuses fonctionnalités et peut donc

se révéler suffisamment puissant pour rendre inutile l'installation et l'utilisation d'un serveur de bases de données externe au langage.

Simplification des tâches courantes

Les autres apports des premières versions de PHP 5 avaient pour objectif de simplifier les tâches les plus courantes. Ainsi, de nombreuses fonctions ont vu le jour et la gestion des erreurs a été repensée. Enfin, la compatibilité avec PHP 4 a été au cœur des préoccupations des développeurs.

PDO : socle commun aux SGBD

PDO (PHP Data Object) a été la principale nouveauté de PHP 5.1. Cette extension apporte un confort d'utilisation et une abstraction plus importante que les anciennes fonctions natives propres à chaque SGBD. L'approche objet de PDO permet par ailleurs d'étendre facilement les fonctions d'accès à votre base de manière transparente.

En interne, PDO permet à l'équipe de développement de PHP de développer beaucoup plus rapidement de nouveaux connecteurs vers de nouvelles bases de données. Au lieu de tout réécrire du début comme auparavant, ils peuvent se baser sur une architecture complète et ne rajouter que ce qui est spécifique.

PDO est un socle commun pour les connecteurs vers les SGBD. Il fournit des fonctions de base et unifie les interfaces utilisateur. Il ne constitue pas à proprement parler un système d'abstraction aux bases de données, bien qu'il puisse servir en ce sens.

Les principaux apports de PHP 5.3

Espaces de nommage

Attendue depuis longtemps, cette fonctionnalité permet d'encapsuler des fonctions et des classes au sein d'espaces spécifiques et ainsi d'éviter les collisions de noms au niveau des constantes, des fonctions, des classes et des interfaces.

PHP Archive

Autrefois disponible sous la forme d'une extension, cette fonctionnalité est maintenant intégrée au langage. PHP 5.3 dispose donc naturellement de tous les outils pour créer des PHAR (abréviation de *PHP Archive*). Similaires aux archives JAR ou WAR du monde Java, les PHAR peuvent contenir un ou plusieurs fichiers, être compressées et sont exécutables directement par PHP.

Pilote natif pour MySQL

Un pilote natif pour le serveur de bases de données MySQL a été intégré à PHP 5.3. Sa mise en œuvre et son utilisation sont complètement transparentes pour le développeur, mais il apporte un gain de performance significatif.

Fonctions anonymes et fermetures lexicales

Les fonctions anonymes, accompagnées des fermetures lexicales, sont l'une des fonctionnalités les plus intéressantes de cette version. Avec PHP 5.3, il est désormais possible de définir simplement une fonction sans nom, et même de l'affecter à une variable, à la manière de ce qui se fait dans d'autres langages, comme JavaScript.

Instruction GOTO

L'instruction `GOTO` a été ajoutée au langage après quelques débats, malgré sa mauvaise réputation. Attention, il est aisé d'écrire du code illisible avec ce motif de programmation.

Syntaxe Nowdoc

La syntaxe `Nowdoc` est aux chaînes de caractères délimitées par des apostrophes ce que la syntaxe `Heredoc` est à celles délimitées par des guillemets. Elle définit simplement une suite de caractères sur plusieurs lignes, où les variables ne seront pas interprétées par PHP.

Résolution statique à la volée

PHP est maintenant capable de calculer lors de l'exécution la classe qui doit être utilisée lors d'un appel à une méthode.

Compatibilité avec PHP 4

Les concepteurs de PHP s'attachent à conserver le plus longtemps possible la compatibilité avec le code existant. Cependant, pour que le langage puisse évoluer, ils ont fini par se résoudre à rendre le moteur de PHP incompatible avec les versions qui ne sont plus officiellement développées, comme les versions 4. En conséquence, PHP 5.3 n'est plus forcément capable d'exécuter du code écrit en PHP 4 sans modification.

Architecture et fonctionnement

Architecture technique

Dans la plupart des déploiements, PHP est utilisé conjointement avec :

- généralement Apache comme serveur HTTP ou, plus rarement, Microsoft IIS ;
- MySQL et Oracle comme SGBD ; on peut aussi rencontrer PostgreSQL ou Microsoft SQL Server ;
- Linux ou BSD comme système d'exploitation ; Windows ou Mac OS sont aussi des possibilités fonctionnelles.

Les plates-formes en production reposent en majorité sur le quatuor Linux, Apache, MySQL et PHP (LAMP).

Grâce à ses nombreux connecteurs et à la prise en charge de Java, COM et .Net, PHP est capable de se connecter à la plupart des applications existantes de l'entreprise. Cette plate-forme peut ensuite exposer l'existant de l'entreprise et les nouveaux développements au travers de différents types d'interfaces :

- web (HTML, WML, etc.) ;
- services web reposant sur Soap ;
- applications graphiques ;
- client riche ;
- Ajax ;
- ligne de commande (CLI) ;
- et même Microsoft Office (Word, Excel), OpenOffice/LibreOffice, Adobe PDF, Macromedia Flash (via Ming), etc.

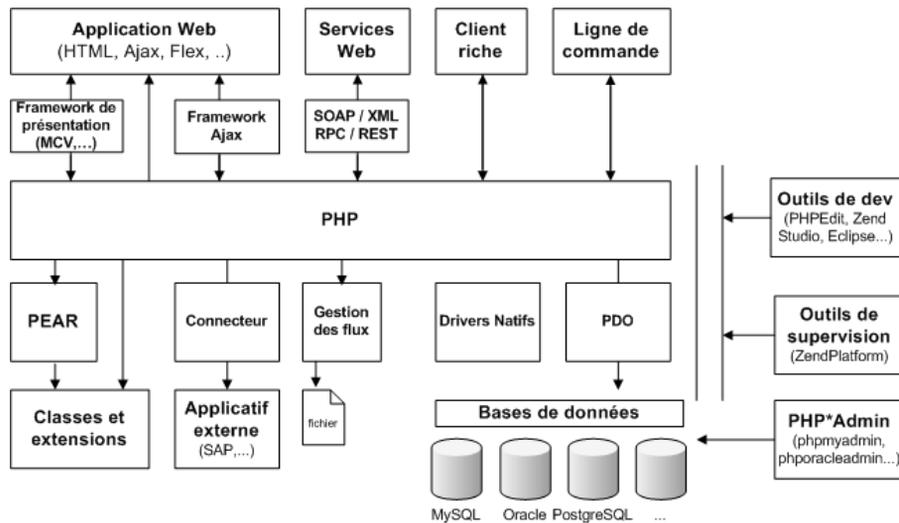


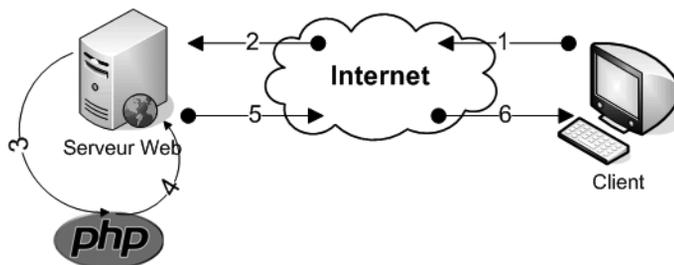
Figure 1-2
Architecture technique de PHP

Fonctionnement de PHP

L'utilisateur qui appelle une page PHP ignore tout du code sous-jacent. Effectivement, ce code est interprété par le serveur avant d'être traduit dans le format de sortie (généralement en HTML, mais aussi en XML, fichier PDF, etc.). Pour ce faire, le serveur web lance l'interpréteur PHP exécutant ainsi le script PHP.

Les commandes figurant dans la page sont interprétées et le résultat prend la forme d'un document publié à la place du code source. À l'issue de cette phase de traduction, la page modifiée est envoyée au client pour y être affichée par le navigateur.

Figure 1-3
Fonctionnement de PHP



Le serveur web reconnaît à l'extension des fichiers, différente de celle des pages HTML simples, si le document appelé par le client comporte du code PHP. L'extension utilisée par les pages PHP peut être définie individuellement dans le fichier de configuration du serveur web. Les extensions courantes pour les pages PHP sont `.php` et `.php5` ; nous utiliserons l'extension `.php` afin d'assurer une compatibilité avec toutes les versions.

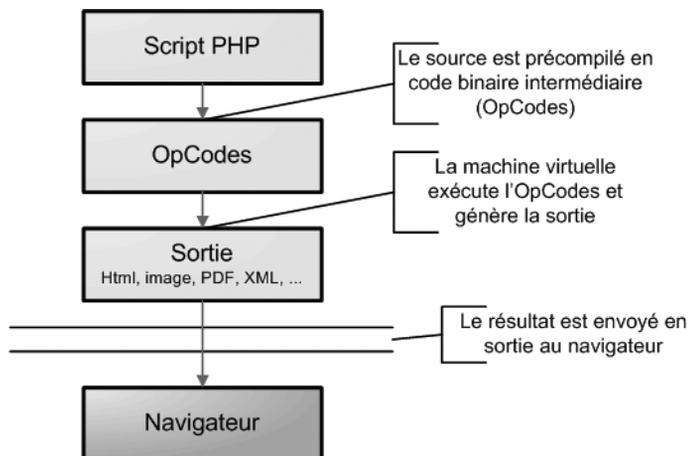
La machine virtuelle de PHP

Le cœur de PHP 5 est basé sur une machine virtuelle. Les concepts sont les mêmes que pour Java et `.Net`. Un précompilateur compile le code source en *byte code* (code intermédiaire), puis l'envoie à la machine virtuelle pour exécution.

Cette architecture permet d'ajouter des outils d'optimisation à l'exécution (cache de code), qui divisent souvent par trois le temps d'affichage d'une page.

PHP 5 propose enfin une API qui permet d'étendre ses fonctionnalités au travers de modules additionnels. Ces modules permettent par exemple de se connecter à une base de données ou à un annuaire LDAP, d'exécuter des composants COM ou Java, de dialoguer en Soap avec des services web, etc.

Figure 1-4
Fonctionnement de la machine virtuelle



PHP en France et dans le monde

LAMP (*Linux Apache MySQL PHP*) est la première plate-forme web dans le monde.

Apache est le serveur le plus utilisé sur Internet avec plus de 60 % de parts de marché, suivi de loin par le serveur IIS de Microsoft, qui totalise aux environs de 18 % de parts de marché (chiffres d'avril 2011, source Netcraft).

On trouve sur le site de PHP des statistiques d'utilisation aux adresses suivantes :

<http://phpadvent.org/2010/usage-statistics-by-ilia-alshanetsky>

<http://blog.pascal-martin.fr/post/statistiques-versions-php-2011-09>

Figure 1-5

Taux d'utilisation de PHP par rapport à d'autres langages

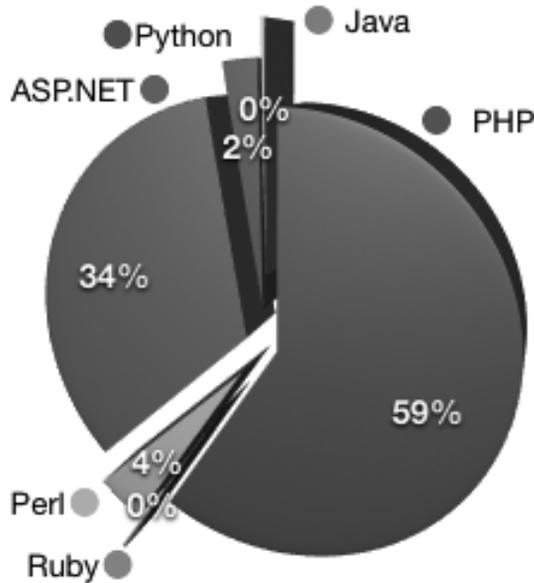
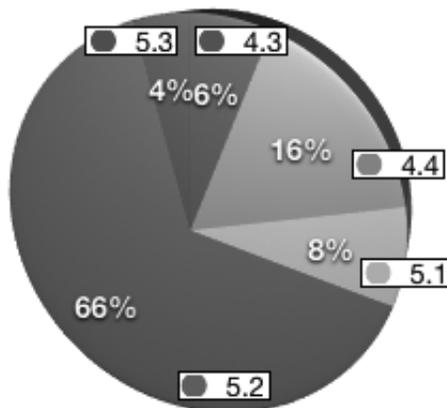


Figure 1-6

Taux d'utilisation des différentes versions de PHP



La communauté française

La France est l'un des acteurs les plus prolifiques sur la scène internationale concernant PHP. Parmi les fers de lance, on compte **Wampserver**, développé par Romain Bourdon, qui permet en quelques clics de souris d'installer Apache, PHP et MySQL sur Windows. Wampserver dispose d'un système d'add-on qui permet, entre autres, de basculer entre différentes versions de PHP en un clic de souris (idéal pour tester vos applications sur différentes versions de PHP). Le logiciel **Dotclear** développé à l'origine par Arnaud Meunier et aujourd'hui maintenu par une équipe de développeurs français, la vieillissante mais toujours utile bibliothèque **FPDF**, permettant de créer du PDF, développée par Olivier Plathey, et **PHPedit**, géré par Sébastien Hordeaux, font aussi partie des références. L'un des frameworks référence, **Symfony**, est également issu du travail du français Fabien Potencier. Emmanuel Faivre, Laurent Abbal et Thierry Murail sont les créateurs d'**Easy-PHP**, un auto-installeur célèbre. N'oublions pas également Vincent Pontier qui est le créateur de la **mascotte de PHP : l'éléphant**.

Figure 1-7

Les principaux
outils français



Outre ces excellents produits, libres d'utilisation, les Français sont très actifs dans de nombreux projets de développement. Ainsi, la France, tout comme l'Allemagne, fait partie des principaux pays impliqués dans le développement de PHP. Les États-Unis, plus axés vers les technologies propriétaires, commencent à s'y mettre mais restent encore peu présents.

Il en résulte de très nombreuses ressources disponibles gracieusement sur Internet. De nombreux bénévoles mettent à disposition des informations sur tous les aspects de PHP. Nous vous proposons de découvrir au travers des pages suivantes les différents sites français composant la communauté PHP en notre pays.

Les ressources d'aide francophones

Il existe de nombreux sites traitant de PHP. Nous avons ici essayé de sélectionner les plus représentatifs malgré la difficulté, tant les sites de qualité sont nombreux.

L'Afup

L'Afup (Association française des utilisateurs de PHP) est une association dont le principal objectif est de promouvoir le langage PHP auprès des professionnels. C'est l'Afup qui organise depuis 2001 le Forum PHP en France (site Internet : <http://www.afup.org>).

Figure 1-8
L'Association française des utilisateurs de PHP



Utilité du site

Vous trouverez de nombreux retours d'expérience, chiffres et conseils sur l'utilisation de PHP. L'objectif est de vous donner les outils pour vendre PHP à vos clients.

Conseil

Inscrivez-vous comme membre et participez au développement et à la promotion du PHP en France.

PHPFrance.com

PHPFrance est l'espace avec lequel de nombreux développeurs PHP d'aujourd'hui se sont formés il y a quelques années. De nos jours, le forum est très actif et peu de questions demeurent longtemps sans réponse. Un espace contenant des cours est extrêmement pratique (site Internet : <http://www.phpfrance.com>).

Figure 1-9
PHPFrance



Index du forum > PHP > PHP avancé

Panneau de l'utilisateur (0 nouveau message privé) • Voir mes messages

[Panneau de modération]

mysql_fetch_assoc et legende d'un graphique

VERROUILLE Dans ce sujet Rechercher

mysql_fetch_assoc et legende d'un graphique

26 Fév 2012, 17:13

Bonjour à tous,
je suis en train d'essayer de récupérer des infos d'une colonne d'un database et les afficher sous forme de "canembar" en utilisant phplot.
j'arrive à passer depuis de mon mysql_fetch_assoc le total pour chaque valeur comme ceci:

Syntaxe: [Télécharger] [Masquer]

```
Code php
$query = "SELECT ID FROMDTTO,COUNT(*) AS col_mmm FROM sortebanque WHERE HESE = 'gennaso' GROUP BY FROMDTTO ASC";
$result = mysql_query($query) or trigger_error(mysql_error()." in ".$query);
while($dbres = mysql_fetch_assoc($result))
{
    $i1 = 0;
    $example_data[] = array($i1,$dbres['col_mmm'],$dbres['FROMDTTO']);
    $i1++;
}
$graph->setDataValues($example_data);
```

Utilité du site

PHPFrance propose de nombreux articles sur l'utilisation de PHP. Vous trouverez également un forum à l'activité débordante où peu de questions restent sans réponse. Accessoirement, un salon IRC (*Internet Relay Chat*) est associé au site : #phpfrance sur le réseau Undernet.

Conseil

Si vous cherchez un développeur PHP ou un emploi sur PHP, allez sur la rubrique nommée « emplois du PHP », vous y trouverez des informations intéressantes.

Consultez le salon IRC #phpfrance sur le réseau Undernet pour retrouver en direct des passionnés de PHP.

Planete-php.fr

Planete-php.fr est un agrégateur des principaux blogs francophones mis en place par l'Association française des utilisateurs de PHP. Ce site permet en un coup d'œil de se tenir au courant des principales nouveautés PHP (site Internet : <http://www.planete-php.fr>).



The screenshot shows the Planete-php.fr website interface. At the top, the site logo 'planete-php.fr' is displayed. Below it, a section titled 'Cyruss blog' features a post about the book 'Performances PHP : dans les bacs !'. The post text includes: 'On touche au bout : nous venons de valider le Bon A Tirer du livre Performances PHP.' Below the text are two book covers for 'Performances PHP: Audit et optimisation LAMP' by Julien Paoli, with authors Baptiste Mieret de Beyer and Guillaume Breaux. To the right of the post, there are links for 'Flux ATOM' and 'Flux RSS'. Below the book covers, there is a paragraph of text starting with 'Au travers de ces expériences et de nos activités professionnelles respectives, nous avons été confrontés à de nombreuses problématiques, dont une récurrente pour toute application Web victime de son succès : l'optimisation des performances. Bien au?il n'y ait pas de solution'. On the right side of the page, under the heading 'Les sources', there is a list of links to various PHP-related resources: Perrick Penet (onpk.net), Maxence Delannoy (Wiip), Frédéric Hardy (mageelbox.net), Nexen.net, Kamelot Blog, Jean François L'épine, AFUP, FoxMask'z h0m3, Stéphane Brun (sbnet), X-HTML.net - L.olic, Mikael Randy, M3Blog, Rom's Blog, and PHP User Group Bordeaux.

Figure 1-10
Planete-php

Utilité du site

En quelques instants, vous serez au courant des principales nouveautés liées au monde PHP. Que ce soit des liens vers des articles, des infos en avant-première ou des réactions, on trouve de tout sur ce site... incontournable !

PHPIndex.com

PHPIndex est l'un des sites pionniers français sur le PHP. Lancé en novembre 1999, ce portail propose de nombreuses ressources et informations sur le PHP. Cet espace s'adresse aux développeurs confirmés qui souhaitent se tenir au courant sur des sujets pointus (site Internet : <http://www.phpindex.com>).

The screenshot shows the PHPIndex.com website. At the top, there is a navigation menu with 'Contenu', 'Recherche', and 'Archives'. The main header features the 'PHPINDEX' logo and the tagline 'LA PASSERELLE FRANÇAISE DES TECHNOLOGIES PHP'. Below the header, there are several sections: a calendar for February 2012, a sidebar with 'À retenir' (containing news about GLOBALIS and PHP4), and a main content area with 'Offres d'emploi' (job listings) and 'Information' (contact details for GLOBALIS).

Figure 1-11
PHPIndex

Utilité du site

Vous trouverez de nombreux liens vers des articles et des cours sur PHP. Les actualités sont intéressantes et généralement orientées professionnels.

Conseil

Si vous cherchez un développeur PHP ou un emploi sur PHP, allez sur la rubrique « jobs », vous y trouverez des informations intéressantes.

Nexen.net

Nexen.net est l'un des plus anciens sites français consacré au PHP. Depuis l'origine, Nexen participe à la réalisation des documentations PHP et MySQL en français : elles

sont disponibles en téléchargement, fréquemment remises à jour, et disposent d'un moteur de recherche perfectionné (site Internet : <http://www.nexen.net>).



Figure 1-12
Nexen

Utilité du site

Les nouvelles vous permettent de suivre les actualités mondiales sur PHP et MySQL. Ces nouvelles sont aussi disponibles sous forme de lettre hebdomadaire. Le système est clair et souvent mis à jour. Une bibliothèque de scripts vous permet également de gagner beaucoup de temps dans la réalisation de vos projets.

Conseil

Inscrivez-vous à la lettre hebdomadaire pour être informé des principales actualités de PHP.

Developpez.com

Developpez.com est un site portail complet sur l'informatique. La partie concernant PHP est riche et ses forums sont très actifs. On trouve de nombreuses sous-catégories concernant les principaux frameworks, bibliothèques et outils sur PHP (site Internet : <http://www.developpez.com>).



Figure 1-13
Developpez.com

PHP Solutions

PHP Solutions est un magazine papier dédié à PHP et MySQL. Il rassemble de nombreux articles intéressants en français. D'origine polonaise, le magazine est traduit dans de nombreuses langues (site Internet : <http://www.phpsolmag.org>).

Figure 1-14
PHP Solutions



Utilité du magazine

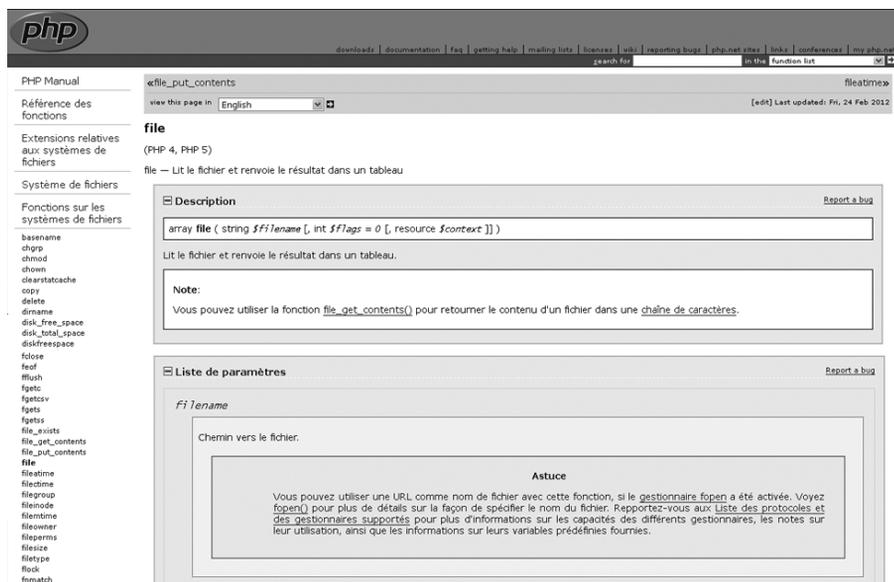
PHP Solutions est désormais gratuit alors il n'y a plus aucune raison de ne pas en profiter.

Les ressources d'aide anglophones

Le site référence PHP

Le site le plus important est le site de PHP lui-même, car il contient la documentation et de nombreuses informations. On notera qu'il existe des miroirs français permettant de disposer d'une bonne rapidité. Le site vous propose automatiquement le plus d'informations possible en français grâce à la détection automatique de votre langue (site Internet : <http://www.php.net>, miroir français : <http://fr.php.net>).

Figure 1-15
Le site PHP.net



Utilité du site

Le site propose un accès à la documentation en ligne. On note également le moteur de recherche des fonctions très utile.

Conseil

Utilisez le moteur de recherche des fonctions. Si vous connaissez le C, indiquez le nom en C de la fonction que vous recherchez. En PHP, son nom est souvent assez proche. Quand vous avez trouvé votre fonction et sa définition comme sur la figure 1-15, consultez les fonctions dans l'espace de gauche, elles concernent toutes le même sujet et peuvent vous permettre de progresser.

MySQL.com

Sur le site de MySQL existe une section dédiée aux développeurs. On y trouve de nombreuses ressources dont des programmes, des articles et des conseils pour optimiser vos applications (site Internet : <http://www.mysql.com>).

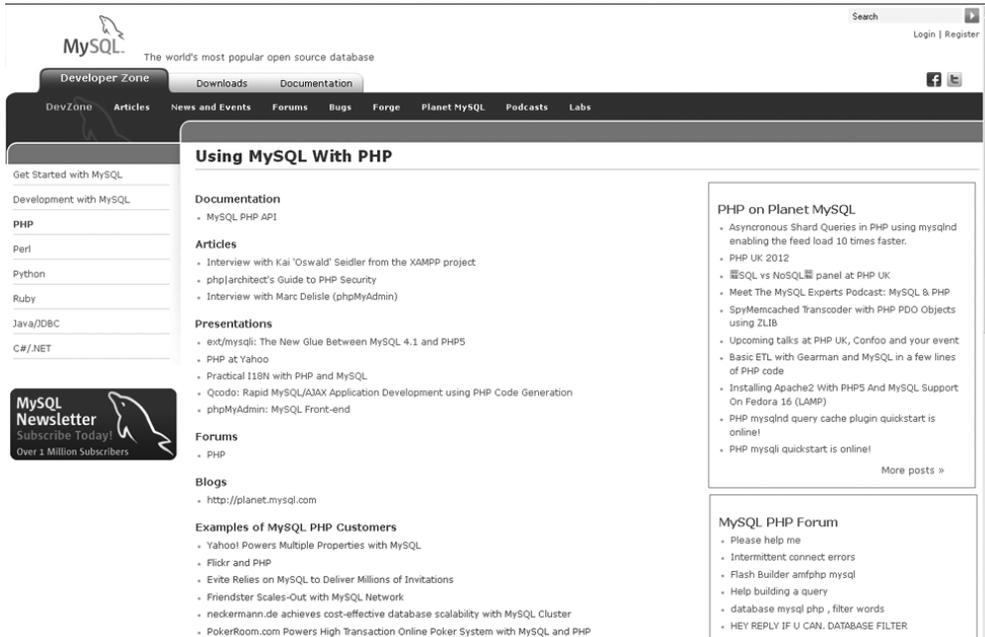


Figure 1-16
MySQL.com

Utilité du site

MySQL.com vous propose de nombreuses ressources pour améliorer vos bases de données. Regardez du côté des programmes proposés pour manipuler vos bases et même pour vous aider à migrer vers MySQL.