

Jean Engels

PHP 7

Cours et exercices



Corrigés des exercices et leur code source

EYROLLES

PHP 7

Un cours idéal pour assimiler la syntaxe et les concepts objet de PHP 7 et s'initier au développement d'applications web professionnelles

Ce manuel d'initiation vous conduira des premiers pas en PHP jusqu'à la réalisation d'un site web complet interagissant avec une base de données MySQL ou SQLite.

Après avoir appris à installer PHP et à créer quelques pages simples, vous étudierez en détail la syntaxe du langage (variables, types de données, opérateurs, instructions, fonctions, tableaux...), avant de progresser rapidement vers des sujets de niveau plus avancé : programmation objet, manipulation des chaînes de caractères et expressions régulières, gestion des mails, sessions et cookies, accès objet aux bases de données MySQL et SQLite, traitements XML, etc.

Successesseur de *PHP 5 – Cours et exercices* du même auteur, cet ouvrage met en avant les nouveautés de PHP 7 : typage des paramètres des fonctions et des valeurs qu'elles retournent, fonctions anonymes, générateurs, classes anonymes, suppression de l'accès procédural à MySQL au profit des méthodes objet, etc.

Des exercices corrigés et des travaux pratiques pour une mise en œuvre immédiate de vos connaissances

Pour vous aider à valider et mettre en œuvre vos connaissances, vous trouverez en fin de chaque chapitre une série d'exercices dont les corrigés et le code source sont disponibles sur www.editions-eyrolles.com et www.funhtml.com. Vous découvrirez également en fin d'ouvrage trois exemples de sites web dynamiques présentés sous forme de travaux pratiques : à vous de développer ces applications à partir du cahier des charges et des indications fournies dans l'énoncé, en résistant à la tentation de télécharger trop rapidement les solutions données sur le site des Éditions Eyrolles !

À qui s'adresse cet ouvrage ?

- Aux étudiants en cursus d'informatique ou de design web.
- À toute personne ayant des bases de programmation web (HTML, JavaScript...) et souhaitant s'autoformer à PHP.
- Aux enseignants et formateurs à la recherche d'une méthode pédagogique pour enseigner PHP.

Enseignant en mathématiques et consultant web, **Jean Engels** est auteur de nombreux ouvrages portant sur les technologies du Web : HTML, CSS, JavaScript, PHP et MySQL.

Sommaire

Premier contact avec PHP • Variables, constantes et types • Les instructions de contrôle (if-else, for, while...) • Les chaînes de caractères et les expressions régulières • Les tableaux • Les formulaires • Les fonctions • Dates et calendriers • La programmation objet (classes et instances, héritage, namespaces...) • Les images dynamiques • La gestion des fichiers • Cookies, sessions et emails • Rappels sur les bases de données relationnelles • Le langage SQL et phpMyAdmin • Accès à MySQL avec mysqli • Accès à MySQL avec la couche d'abstraction PDO • La base de données SQLite • PHP et SimpleXML • Travaux dirigés : site de rencontres, dictionnaire de citations interactif, site de commerce électronique.

PHP 7

AUX ÉDITIONS EYROLLES

Du même auteur

J. ENGELS. – **HTML 5 et CSS 3. Cours et exercices.**

N°13400, 2012, 550 pages.

Autres ouvrages sur PHP et ses frameworks

P. MARTIN, J. PAULI, C. PIERRE DE GEYER et E. DASPET. – **PHP 7 avancé.**

N°14357, 2016, 728 pages.

M. CHAVELLI. – **Découvrez le framework PHP Laravel.**

N°14398, collection OpenClassrooms, 2016, 326 pages.

A. BACCO, préface de F. POTENCIER. – **Développez votre site web avec le framework Symfony3.**

N°14403, collection OpenClassrooms, 2016, 536 pages.

C. CAMIN. – **Développer avec Symfony2.**

N°14131, 2015, 450 pages.

Dans la même collection

G. SWINNEN. – **Programmer avec MySQL.**

N°14302, 4^e édition, 2015, 480 pages.

C. DELANNOY. – **Programmer en Java.**

N°11889, 9^e édition, 2014, 940 pages (réédition avec nouvelle présentation, 2016).

G. SWINNEN. – **Apprendre à programmer avec Python 3.**

N°13434, 3^e édition, 2012, 435 pages.

P. ROQUES. – **UML 2 par la pratique.**

N°12565, 7^e édition, 2009, 396 pages.

G. SWINNEN. – **Apprendre à programmer avec Python 3.**

N°13434, 3^e édition, 2012, 435 pages.

P. ROQUES. – **UML 2 par la pratique.**

N°12565, 7^e édition, 2009, 396 pages.

Jean Engels

PHP 7

Cours et exercices

© Groupe Eyrolles, 2017, ISBN : 978-2-212-67360-9.
61, bd Saint-Germain
75240 Paris Cedex 05
www.editions-eyrolles.com

En application de la loi du 11 mars 1957, il est interdit de reproduire intégralement ou partiellement le présent ouvrage, sur quelque support que ce soit, sans l'autorisation de l'Éditeur ou du Centre français d'exploitation du droit de copie, 20, rue des Grands-Augustins, 75006 Paris.

Table des matières

Avant-propos	XIX
CHAPITRE 1	
Introduction	1
Avant de commencer	2
Compétences requises	2
Installation d'un serveur local	3
Premier contact avec PHP	5
Organisation de PHP	5
Structure des fichiers HTML	6
Écriture du code PHP	10
Ajout de commentaires	14
CHAPITRE 2	
Variables, constantes et types	15
Les variables	15
Affectation par valeur et par référence	16
Les variables prédéfinies	18
Les opérateurs d'affectation combinée	19
Les constantes	20
Définir ses constantes personnalisées	20
Les constantes prédéfinies	22
Les types de données	22

Déterminer le type d'une variable	23
La conversion de type	24
Contrôler l'état d'une variable	25
Les entiers	26
Les flottants	27
Les opérateurs numériques	28
Les fonctions mathématiques	29
Les booléens	31
Le type boolean	31
Les opérateurs booléens	32
Les chaînes de caractères	34
Définir des chaînes	34
Concaténer des chaînes	36
Les tableaux	36
Les objets	40
Les types divers	41
Le type resource	41
Le type NULL	42
Mémo des fonctions	42
Exercices	43

CHAPITRE 3

Les instructions de contrôle	45
Les instructions conditionnelles	45
L'instruction if	45
L'instruction if...else	46
Les opérateurs ? et ??	49
L'instruction switch...case	50
Les instructions de boucle	52
La boucle for	52
La boucle while	55
La boucle do...while	56

La boucle foreach	57
Sortie anticipée des boucles	59
Gestion des erreurs	62
Suppression des messages d'erreur	63
Gestion des exceptions	64
Exercices	69
CHAPITRE 4	
Les chaînes de caractères	71
Affichage des chaînes	71
Affichage formaté	72
Longueur d'une chaîne et codes des caractères	75
Mise en forme des chaînes	76
Modification de la casse	76
Gestion des espaces	78
Entités HTML et caractères spéciaux	79
Recherche de sous-chaînes	81
Comparaison de chaînes	84
Transformation de chaînes en tableaux	87
Les expressions régulières	88
Définition d'un motif élémentaire	88
Les fonctions de recherche PHP	93
Définition d'un motif complexe	95
Mémo des fonctions	97
Exercices	100
CHAPITRE 5	
Les tableaux	103
Créer des tableaux	103
La fonction array()	103
Créer des suites	108

Créer un tableau à partir d'une chaîne	109
Compter le nombre de valeurs d'un tableau	110
Lire les éléments des tableaux	112
Lire avec une boucle for	112
Lire avec une boucle while	114
Lire à l'aide de la fonction each()	116
Lire avec each() et list()	119
L'instruction foreach	121
Manipuler des tableaux	124
Extraire une partie d'un tableau	124
Ajouter et enlever des éléments	127
Opérations sur plusieurs tableaux	130
Trier les éléments d'un tableau	133
Trier des tableaux indicés	134
Trier des tableaux associatifs	139
Opérer une sélection des éléments	142
Appliquer une fonction à un tableau	143
L'objet ArrayObject	146
Création d'un objet tableau	146
Les méthodes de tri des éléments	149
Mémo des fonctions	153
Exercices	158

CHAPITRE 6

Les formulaires	159
Création d'un formulaire HTML	159
L'élément <input />	161
L'élément <textarea>	165
L'élément <select>	166
Exemple de code <form>	166
Récupération des données du formulaire	169
Valeurs uniques	169

Les valeurs multiples	177
Transfert de fichiers vers le serveur	180
Transfert de plusieurs fichiers	184
Gérer les boutons d'envoi multiples	187
Exercices	189
CHAPITRE 7	
Les fonctions	191
Les fonctions natives de PHP	191
Créer ses propres fonctions	193
Définir une fonction	194
Les fonctions qui ne retournent pas de valeur	196
Typage des paramètres	198
Les fonctions qui retournent une valeur	200
Retourner plusieurs valeurs	202
Les paramètres par défaut	203
Les fonctions avec un nombre de paramètres variable	205
Les paramètres de type array	205
Les fonctions particulières de PHP	206
L'opérateur	208
Les générateurs	209
Création d'un générateur	209
Délégation d'un générateur	211
Portée des variables	211
Variables locales et globales	212
Les variables statiques	214
Passer des arguments par référence	216
Cas particuliers	218
Les fonctions dynamiques	218
Les fonctions conditionnelles	220
Fonction définie dans une autre fonction	221
Les fonctions récursives	222

Les fonctions anonymes (closure)	224
Exercices	227
CHAPITRE 8	
Dates et calendriers	229
Les dates	229
Définir une date	232
Vérifier une date	233
Afficher une date en clair	234
La fonction getdate()	237
Afficher la date en français	237
Les fonctions de calendrier	241
Mémo des fonctions	244
Exercices	244
CHAPITRE 9	
La programmation objet	247
Terminologie des objets	248
Classe et instance	249
Création d'une classe	249
Créer un objet	254
Accès aux variables de la classe	257
Les modificateurs d'accessibilité	260
Propriétés et méthodes statiques	262
Constructeur et destructeur d'objet	264
Déréférencement	267
Héritage	268
Enrichir un objet	269
Création d'une classe dérivée	270
Les traits	272
Late Static Binding	279
Les classes abstraites	280

Les interfaces	282
Méthode et classe finales	284
Les classes anonymes	285
Clonage d'objet	287
Les namespaces	289
Création et utilisation	289
Utilisation des alias	294
Méthodes magiques	295
Mémo des fonctions	297
Exercices	297
CHAPITRE 10	
Les images dynamiques	299
Principes généraux	299
Création du cadre de l'image	301
Création des couleurs	303
Tracé de formes géométriques	304
Écriture de texte	313
Utilisation pratique	315
Mémo des fonctions	320
Exercices	323
CHAPITRE 11	
Les fichiers	325
Création, ouverture et fermeture d'un fichier	326
Ouverture du fichier	326
Fermeture du fichier	328
Verrouillage des fichiers	329
Écriture dans un fichier	330
Conserver une information	330
Formatage des données	332

Lecture de fichiers	335
Lire une ligne à la fois	335
Lire un nombre de caractères donné	337
Lire un caractère à la fois	341
Lecture d'une partie d'un fichier	342
Lecture de données formatées	343
Lecture de la totalité d'un fichier	346
Modifications de fichiers	350
Copier un fichier	350
Renommer un fichier	351
Effacer un fichier	352
Informations sur les fichiers	352
Existence d'un fichier	352
Taille des fichiers	353
Mémo des fonctions	355
Exercices	357

CHAPITRE 12

Cookies, sessions et e-mails	359
Les cookies	359
Écriture des cookies	360
Lecture des cookies	362
Exemple de page avec cookies	363
Les sessions	366
Le mécanisme des sessions	367
Session avec cookie	367
La gestion de panier	372
Les sessions sans cookie	377
L'envoi d'e-mails	378
La fonction mail()	378
Envoi d'e-mails au format texte	379
Envoi d'e-mails au format HTML	382
Mémo des fonctions	385

Exercices	386
CHAPITRE 13	
Rappels sur les SGBDR	387
Le modèle entité/association	388
Les entités	388
Les attributs	389
Les associations	390
Les cardinalités	390
Conception du MCD	393
Normalisation du MCD	393
La base magasin en ligne	394
Passage au modèle relationnel	395
Le modèle relationnel	395
Conception du MLD	396
Le MLD de la base magasin en ligne	398
Modèle physique de données	398
Exercices	399
CHAPITRE 14	
Le langage SQL et phpMyAdmin	401
L'interface phpMyAdmin	401
Création d'une base de données	403
Création de tables	405
Les types de données MySQL	405
Création des tables	409
Modification des tables	414
Insertion de données	419
Insertion ligne par ligne	419
Mise à jour des données	420
Importation à partir d'un fichier texte	421
Insertion à partir d'un fichier Excel	423

Les données de la base magasin	424
Sélection des données	426
Sélection dans une table	426
Les jointures	439
Jointure de deux tables	439
Jointure de plus de deux tables	441
Exercices	442

CHAPITRE 15

Accès objet à MySQL avec PHP	445
Connexion au serveur MySQL	445
Envoi de requêtes SQL au serveur	448
Lecture du résultat d'une requête	449
Lecture à l'aide d'un tableau	449
Lecture des noms de colonnes	452
Récupération des valeurs dans un objet	454
Insertion de données dans la base	457
Insertion des données	457
Mise à jour d'une table	460
Recherche dans la base	464
Les requêtes préparées	467
Les transactions	470
Mémo des méthodes et propriétés	472
Classe mysqli : méthodes	472
Classe mysqli : propriétés	473
Classe mysqli_result : méthodes	473
Classe mysqli_result : propriétés	474
Classe mysqli_stmt : méthodes	474
Classe mysqli_stmt : propriétés	475
Exercices	475

CHAPITRE 16	
PDO et MySQL	477
Connexion au serveur MySQL	478
Envoi de requêtes SQL au serveur	480
Lecture du résultat d'une requête	481
Lecture à l'aide d'un tableau	482
Lecture des noms de colonnes	483
Récupération des valeurs dans un objet	486
Insertion de données dans la base	489
Insertion des données	489
Mise à jour d'une table	492
Recherche dans la base	496
Les requêtes préparées	499
Les transactions	502
Mémo des méthodes	504
Classe PDO	504
Classe PDOStatement	505
Classe PDOException	505
Exercices	506
CHAPITRE 17	
La base SQLite	507
Caractéristiques générales	507
L'interface SQLiteManager	509
Méthodes d'accès à SQLite	511
La méthode SQLite3	512
Connexion à la base	512
Envoi des requêtes	513
Insertion de données	515
Les transactions	517
Lecture des résultats d'une requête	519

Création de fonctions SQL personnalisées	521
Les requêtes préparées	523
Accès à SQLite avec PDO	525
Mémo des méthodes des objets	526
Classe SQLite3	526
Classe SQLite3Result	527
Classe SQLite3Stmt	528
Exercices	528
CHAPITRE 18	
PHP et SimpleXML	531
Notions de XML	531
Lecture d'un fichier XML	535
Accéder au contenu d'un fichier XML	535
Lecture des attributs d'un élément	539
Lecture d'un fichier à structure complexe	541
Modification des valeurs des éléments et des attributs	543
Recherche dans un fichier	544
Création d'un fichier XML à partir d'un formulaire	548
Relations entre XML et une base MySQL	550
Création d'un fichier XML à partir d'une table MySQL	550
Création d'une table MySQL à partir d'un fichier XML	556
Mémo des fonctions et méthodes	558
Exercices	558
CHAPITRE 19	
Travaux personnels	561
Démarche à suivre	561
TP n° 1. Un site de rencontres	562
L'interface	562
La base de données SQLite	565

TP n° 2. Dictionnaire de citations interactif	565
L'interface	566
La base de données MySQL	567
TP n° 3. Commerce en ligne	568
Les besoins du client	568
Votre travail	569
Index	571

Avant-propos

Cet ouvrage est destiné en priorité à ceux qui veulent se former à PHP et aux bases de données MySQL et SQLite pour créer des pages web dynamiques et interactives. Nous y présentons à la fois les bases du langage, qui étaient déjà celles de PHP 5, et les nouveautés de la version 7. Disons-le d'emblée, la version 7 ne représente pas un saut conceptuel comme l'ont été les différentes versions 5 par rapport à PHP 4.

Nous pouvons néanmoins noter entre autres des évolutions dans la rigueur du langage avec, par exemple, les typages des paramètres des fonctions et des valeurs qu'elles retournent, la création des fonctions anonymes, des générateurs, l'ajout de quelques opérateurs et fonctions utiles et, dans le domaine des objets, l'ajout des classes anonymes.

De plus, dans la gestion de MySQL, l'extension `mysql` qui permettait un accès procédural a été supprimée au profit d'un accès objet à MySQL – ce qui confirme la tendance déjà bien amorcée par PHP 5 de la généralisation de l'utilisation des objets dans tous les domaines.

Les concepteurs de PHP 7 confirment ainsi vouloir en faire un langage encore plus professionnel et solide, tout en conservant la facilité d'accès et l'efficacité qui ont fait son immense succès.

Les exercices proposés à la fin de chaque chapitre vous offriront l'occasion de mettre en œuvre immédiatement les points étudiés. Les travaux personnels proposés à la fin de l'ouvrage vous permettront de mettre en œuvre l'ensemble des connaissances acquises dans des cas réels de sites web dynamiques.

Vous pourrez mesurer votre compréhension des notions abordées grâce aux corrigés de ces exercices, téléchargeables sur le site www.editions-eyrolles.com, ainsi que visibles et exécutables sur le site www.funhtml.com.

L'ouvrage est divisé en dix-neuf chapitres, qui abordent successivement les sujets suivants.

- Le chapitre 1 rappelle le fonctionnement général de PHP dans la création de pages dynamiques. Il montre comment installer les outils nécessaires aux tests des scripts, en particulier le serveur web Apache.PHP.MySQL.

- Le chapitre 2 définit les différents types de données manipulables avec PHP et montre comment les utiliser en créant des variables ou des constantes.
- Le chapitre 3 fait un tour d'horizon des instructions de contrôle indispensables à tout langage. Il montre comment créer des instructions conditionnelles et des boucles ainsi que gérer les erreurs par le mécanisme des exceptions PHP.
- Le chapitre 4 traite de la création et de la manipulation des chaînes de caractères. Il décrit les différentes techniques d'affichage, simple ou formaté, des chaînes et présente l'écriture d'expressions régulières.
- Le chapitre 5 se penche sur la création de tableaux, un type de données très pratique aux multiples applications. Diverses techniques de lecture des tableaux sont explicitées à l'aide de nombreux exemples ainsi qu'un accès objet aux tableaux.
- Le chapitre 6 détaille la création des formulaires, qui sont les vecteurs indispensables au transfert d'informations entre le poste client et le serveur, et est désormais entièrement adapté à HTML 5 qui a ajouté un grand nombre de composants utiles. Il montre comment récupérer et gérer les données saisies par les visiteurs d'un site.
- Le chapitre 7 est consacré aux fonctions qui permettent une meilleure organisation des scripts. Le passage d'arguments par valeur et par référence ainsi que la gestion des paramètres et le retour des valeurs multiples par une fonction y sont détaillés. S'y ajoutent des nouveautés comme le typage des paramètres et des valeurs retournées ainsi que les fonctions anonymes et les générateurs.
- Le chapitre 8 fait le tour des outils permettant le calcul des durées et la gestion des dates et des calendriers avec PHP.
- Le chapitre 9 aborde le modèle objet de PHP et introduit les méthodes qui révolutionnent la création d'objets avec PHP, le rapprochant ainsi des langages de POO. L'ajout des traits a fait faire un pas supplémentaire dans la modularisation du code tout comme des transformations pour la gestion des namespaces. Une autre innovation comme les fonctions anonymes est introduite.
- Le chapitre 10 montre comment PHP est capable de créer des images dynamiques susceptibles de rendre les sites plus attractifs par la création d'images GIF, JPEG ou PNG en fonction des besoins et aussi à partir de données. Quelques nouvelles fonctions de modification des images sont ajoutées dans la version 7.
- Le chapitre 11 aborde la gestion des fichiers sur le serveur et livre une première approche du stockage sur le serveur d'informations issues du poste client. Les différentes méthodes de création de fichiers, de lecture et d'écriture de données y sont décrites en détail.
- Le chapitre 12 est dédié à la création et à la gestion des cookies ainsi qu'au mécanisme des sessions, qui permet la conservation et la transmission d'informations entre toutes les pages d'un même site. La création et l'envoi d'e-mails pour renforcer les possibilités de contact entre l'internaute et le site sont également abordés.

- Le chapitre 13 rappelle les notions théoriques indispensables à la modélisation d'une base de données. Il dresse une rapide synthèse du modèle entité/association et du passage au modèle relationnel, qui est utilisé par la plupart des SGBD actuels, en particulier MySQL et SQLite, qui font l'objet des chapitres suivants.
- Le chapitre 14 est un rappel du langage SQL en vue de son utilisation dans MySQL. Ce survol est réalisé en dehors du contexte PHP au moyen de l'interface de gestion phpMyAdmin.
- Le chapitre 15 explique comment accéder à une base MySQL au moyen de scripts PHP en utilisant l'extension mysqli, qui permet l'accès à MySQL uniquement en POO, l'accès procédural étant maintenant obsolète. Y sont abordées les différentes commandes d'insertion et de mise à jour de données ainsi que de lecture et de recherches élaborées sur une ou plusieurs tables au moyen de jointures.
- Le chapitre 16 présente la couche d'abstraction PDO qui permet l'accès à MySQL mais également à d'autres bases de données et qui représente une solution d'avenir dans ce domaine.
- Le chapitre 17 aborde la base de données embarquée SQLite, introduite lors de la parution de PHP 5. Comme pour MySQL, l'accès procédural à SQLite est supprimé et la méthode objet s'impose.
- Le chapitre 18 dévoile SimpleXML qui permet la manipulation en lecture et en écriture des fichiers XML, d'une manière qui est nettement simplifiée par rapport à celle de la version précédente.
- En conclusion, le chapitre 19 est constitué de trois sujets de travaux personnels, que vous devrez réaliser en faisant appel aux connaissances acquises tout au long des chapitres précédents. De difficulté croissante, ces sujets vous permettront d'évaluer de manière concrète la pertinence de vos acquisitions. Les corrigés de ces travaux personnels sont donnés et utilisables sur le site : <http://www.funhtml.com>.

1

Introduction

Le sigle PHP signifiait à l'origine *Personal Home Page*. Pour Rasmus Lerdorf, l'auteur de ce qui allait devenir le langage de script côté serveur incorporable dans tout document HTML que nous connaissons, il s'agissait alors d'ajouter quelques fonctionnalités à ses pages personnelles. PHP signifie aujourd'hui *Php Hypertext Preprocessor* car il renvoie à un navigateur un document HTML construit par le moteur de script Zend Engine 2 de PHP, dont nous allons voir le fonctionnement. Il permet de créer des pages web dynamiques et interactives.

Imaginez que vous soyez fan de moto et que vous vouliez présenter les photos de vos modèles préférés et leurs caractéristiques techniques. La création de quelques pages HTML statiques, agrémentées de liens pour naviguer d'une page à l'autre, peut suffire. Imaginez maintenant que vous soyez rejoint par d'autres personnes qui partagent la même passion et que votre site présente des centaines de modèles et une rubrique de petites annonces et de contacts entre membres. La quantité d'informations à présenter ne permet plus de naviguer dans le site au moyen de liens mais réclame, dès la page d'accueil, un moteur de recherche. L'utilisateur saisit un ou plusieurs critères de recherche, à partir desquels le code d'un script PHP crée une page contenant les informations recherchées et seulement elles. Chaque visiteur et chaque besoin particulier génèrent donc des pages différentes, personnalisées, construites dynamiquement.

PHP permet en outre de créer des pages interactives. Une page interactive permet à un visiteur de saisir des données personnelles. Ces dernières sont ensuite transmises au serveur, où elles peuvent rester stockées dans une base de données pour être diffusées vers d'autres utilisateurs. Un visiteur peut, par exemple, s'enregistrer et retrouver une page adaptée à ses besoins lors d'une visite ultérieure. Il peut aussi envoyer des e-mails et des fichiers sans avoir à passer par son logiciel de messagerie. En associant toutes ces

caractéristiques, il est possible de créer aussi bien des sites de diffusion et de collecte d'information que des sites d'e-commerce, de rencontres ou des blogs.

Pour contenir la masse d'informations collectées, PHP s'appuie généralement sur une base de données, généralement MySQL mais aussi SQLite, et sur des serveurs Apache. PHP, MySQL et Apache forment d'ailleurs le trio ultradominant sur les serveurs Internet. Quand ce trio est associé sur un serveur à Linux, on parle de système LAMP (Linux, Apache, MySQL, PHP). PHP est utilisé aujourd'hui par plus des trois quarts des sites dynamiques de la planète et par les trois quarts des grandes entreprises françaises. Pour un serveur Windows, on parle de système WAMP, mais ceci est beaucoup moins courant.

Vous passerez en revue dans le cours de cet ouvrage tous les outils nécessaires à la réalisation d'un site dynamique et interactif à l'aide de PHP et d'une base de données MySQL ou SQLite. Les principaux avantages de ces outils sont la facilité d'apprentissage, la grande souplesse d'utilisation, l'excellent niveau de performance et, ce qui ne gêne rien, la gratuité.

Pour parvenir à la réalisation des types de site que nous venons de voir nous allons aborder successivement les points suivants :

- La syntaxe et les caractéristiques du langage PHP, dont la connaissance est la base indispensable à toute la suite.
- Les notions essentielles du langage SQL permettant la création et la gestion des bases de données et la réalisation des requêtes sur ces bases.
- Le fonctionnement et la réalisation de bases de données MySQL puis SQLite et les moyens d'y accéder à l'aide des fonctions spécialisées de PHP ou d'objets.

Pour progresser rapidement il vous sera nécessaire de lire ce livre de manière linéaire au moins pour le début et de ne pas brûler les étapes. N'essayez donc pas de commencer par la fin en abordant les bases de données sans connaissance préalable de PHP ou de SQL.

Avant de commencer

Avant d'envisager d'écrire votre premier script, il vous faut faire le point sur les connaissances nécessaires à cette réalisation. Il n'est pas envisageable de commencer cet apprentissage sans aucune connaissance d'Internet et de la création de pages HTML. Du point de vue matériel, vous devez de surcroît disposer des quelques outils qui vous permettront d'écrire et surtout de tester vos scripts sur un ordinateur personnel.

Compétences requises

L'objectif de cet ouvrage étant de permettre un apprentissage progressif de PHP, la connaissance d'un langage de programmation quelconque n'est pas vraiment indispensable. Cependant, quelques notions de programmation en langage C, Java ou en JavaScript, par exemple, ne peuvent que rendre l'accès à PHP plus facile. En revanche, la connaissance du langage HTML est recommandée puisque le serveur PHP renvoie les pages HTML que vous programmez.

Pour ce qui concerne la méthode, commencez par télécharger et tester les exemples du livre, puis modifiez-en certains paramètres afin d'évaluer le rôle de chacun d'eux. Cela vous permettra de mieux apprécier l'effet réel d'une instruction, par exemple.

Les outils de création

Puisqu'il s'agit de construire des pages web et de produire un document HTML lisible par un navigateur, un éditeur HTML peut convenir pour créer la structure générale des pages, y compris s'il est WYSIWYG. Le code des scripts PHP peut quant à lui être écrit dans n'importe quel éditeur de texte, tel que le Bloc-notes de Windows.

Si les éditeurs tels que Dreamweaver privilégient l'aspect visuel en cachant le code, d'autres outils de création très simples, comme HTML Kit, obligent le programmeur à voir en permanence les éléments HTML utilisés. Un bon compromis consiste à utiliser un éditeur WYSIWYG pour créer le design et la mise en page générale des pages web puis de récupérer le fichier HTML réalisé dans un éditeur PHP spécialisé afin d'effectuer les tests facilement après avoir installé le serveur local PHP.

Le tableau 1-1 présente une liste d'outils de développement de scripts.

Tableau 1-1 – Éditeurs HTML et PHP

Produit	Statut	Description	Adresse
HTML Kit	Gratuit	Éditeur HTML	http://www.htmlkit.com
HTMLPad Rapid PHP	Payants	Éditeurs HTML et PHP	http://www.blumentals.net
EditPlus	Payants	Éditeur HTML permettant l'écriture et l'exécution de scripts PHP	http://www.editplus.com
Zend Studio	Payants	Très complet	http://www.zend.com

Installation d'un serveur local

Faute de disposer d'un serveur local sur votre ordinateur personnel, vous seriez obligé pour tester vos pages PHP de les transférer sur le serveur distant de votre hébergeur puis d'appeler ces pages en vous connectant au site à l'aide de votre navigateur. La moindre erreur de code ou la moindre modification vous obligerait à répéter toute cette procédure, d'où une importante perte de temps.

Il est donc indispensable d'installer sur votre poste de travail un serveur local simulant votre serveur distant et vous permettant d'effectuer en direct tous les tests désirés. Vous aurez alors dans votre navigateur exactement le même aspect pour toutes ces pages que les visiteurs de votre site quand vous aurez opéré le transfert de vos fichiers sur le serveur distant qui l'hébergera.

Le serveur local comprend les éléments suivants, disponibles séparément aux adresses entre parenthèses :

- Serveur Apache (<http://www.apache.org>).
- Interpréteur de code PHP (<http://www.php.net>).
- Base de données MySQL (<http://www.mysql.com>).
- Base de données SQLite (<http://www.sqlite.org>).
- Utilitaire phpMyAdmin, qui permet de créer et de gérer bases et tables de données MySQL (<http://www.phpmyadmin.net>).
- Utilitaire SQLiteManager, qui permet de créer et de gérer bases et tables de données SQLite (<http://www.sqlitemanager.org>).

On peut trouver sur le Web divers packages complets pour Windows, Linux ou Mac, qui permettent d'installer en une seule opération tous ces éléments, évitant du même coup les problèmes de configuration.

Un installateur est apparu à l'occasion de la sortie de PHP 5. Son auteur, Romain Bourdon, se montre très réactif en publiant une nouvelle version à chaque évolution. Son package, nommé Wampserver, téléchargeable à l'adresse <http://www.wampserver.com>, est destiné aux ordinateurs sous Windows.

Une fois la procédure de téléchargement terminée, il vous suffit de lancer l'exécutable, qui installe automatiquement Apache, PHP, MySQL, SQLite phpMyAdmin et SQLite-Manager sur votre ordinateur. Si, pendant la phase d'installation, vous avez choisi d'installer PHP en tant que service Windows, le serveur est lancé automatiquement à chaque démarrage du système d'exploitation.

Sous Mac il existe un très bon installateur de serveur local nommé MAMP, très facile d'emploi – c'est celui-ci que nous présentons ici mais WampServer est aussi accessible.

Pour pouvoir être exécutés par le serveur local, tous les scripts que vous écrivez doivent être enregistrés dans le sous-dossier `www` sous Windows ou `MAMP/htdocs` sous Mac. Dans ce dernier, vous pouvez créer un ou plusieurs sous-dossiers correspondant à chaque site que vous voulez tester.

La page de démarrage du serveur vous permet de lancer les serveurs Apache et MySQL (voir figure 1-1). L'icône « Ouvrir la page Webstart » déclenche l'ouverture de la page d'accueil dans votre navigateur par défaut. Cette page d'administration du serveur local (voir figure 1-2) vous donne accès à différents paramètres, l'onglet « Mon site web » affiche la liste de tous les fichiers présents dans le dossier `htdocs` et l'onglet « Outils » donne

accès à phpMyAdmin pour gérer les bases MySQL et à phpLiteAdmin pour les bases SQLite.



Figure 1-1

La page de démarrage des serveurs



Figure 1-2

Page d'administration du serveur local Apache PHP MySQL

Premier contact avec PHP

Étant désormais doté de tous les outils nécessaires, vous pouvez aborder le fonctionnement de PHP et les différentes méthodes de travail que vous devrez utiliser par la suite.

Organisation de PHP

PHP ne repose pas sur une hiérarchie de classes regroupées en sous-ensembles (namespace), comme ASP.Net ou Java, mais sur des modules. Le module de base, dit standard, permet d'accéder aux instructions élémentaires, aux différents types de données et à un grand nombre de fonctions. Des modules additionnels spécialisés permettent d'ajouter des fonctionnalités particulières, comme l'accès aux diverses bases de données et leur gestion. Chaque module donne accès à un grand nombre de fonctions spécialisées pour un domaine particulier.

La liste des modules disponibles actuellement est visible dans la documentation générale du langage sur le site officiel de PHP, à l'adresse <http://www.php.net>.

Vous pouvez télécharger sur le même site la documentation officielle de PHP, qui donne, y compris en français, la définition de toutes les fonctions existantes. Le document compte quelque deux mille pages au format Acrobat PDF.

Pour savoir quels modules vous pouvez utiliser sur votre serveur local, il vous suffit de cliquer sur l'onglet `phpinfo` de la page d'accueil.

Pour obtenir la même information pour le serveur qui héberge votre site, procédez de la façon suivante :

1. Écrivez le script PHP suivant, d'une simplicité enfantine (vous n'en écrirez jamais d'aussi court donnant autant d'informations), à l'aide de l'éditeur que vous avez choisi :

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

2. Enregistrez le script sous le nom `info.php`. Sous PHP, tous les scripts commencent par la ligne `<?php` et se terminent par `?>`. Notez que, sauf recommandation spéciale de votre hébergeur, tous les fichiers qui contiennent des instructions PHP sont enregistrés avec l'extension `.php`. Les extensions `.php3`, `.php4`, `.php5` ou `.phtml` se rencontrent sur certains serveurs, suivant la configuration effectuée par l'administrateur.
3. Transférez le fichier `info.php` sur votre serveur distant à l'aide d'un logiciel FTP. Si vous n'en avez pas, vous pouvez télécharger FileZilla, un logiciel gratuit, dont le fonctionnement est aussi simple que convivial, à l'adresse <http://www.sourceforge.net/projects/filezilla>.
4. Saisissez l'adresse <http://www.votresite.com/info.php> dans votre navigateur.

Un grand nombre d'informations utiles concernant votre serveur et l'ensemble des modules qui y sont installés apparaissent alors sur toute la hauteur de la page (voir figure 1-3).

Il est recommandé d'imprimer ces informations et de les conserver précieusement car elles vous permettront de déterminer, au moment où vous en aurez besoin, si vous pouvez utiliser tel ou tel module ou fonction. Il serait dommage de travailler des heures à créer un script qui utilise des fonctions utilisables en local mais non disponibles sur votre serveur distant.

Structure des fichiers HTML

Comme expliqué précédemment, la connaissance du langage HTML est utile pour se lancer dans l'écriture de scripts PHP. Il est donc utile de connaître la structure des fichiers HTML car une page dynamique PHP est bien un document HTML envoyé par le serveur vers le poste client.

PHP Version 7.0.0	
System	Darwin Air-de-ENGELS 16.3.0 Darwin Kernel Version 16.3.0: Thu Nov 17 20:23:58 PST 2016; root:xnu-3789.31.2~1/RELEASE_ARM_T8020
Build Date	Dec 8 2015 16:35:17
Configure Command	./configure '--with-apxs2=/Applications/MAMP/Library/bin/apxs' '--with-gd' '--with-jpeg-dir=/Applications/MAMP/Library' '--with-png-dir=/Applications/MAMP/Library' '--with-zlib' '--with-zlib-dir=/Applications/MAMP/Library' '--with-freetype-dir=/Applications/MAMP/Library' '--prefix=/Applications/MAMP/bin/php/php7.0.0' '--exec-prefix=/Applications/MAMP/bin/php/php7.0.0' '--sysconfdir=/Applications/MAMP/bin/php/php7.0.0/conf' '--with-config-file-path=/Applications/MAMP/bin/php/php7.0.0/conf' '--enable-ftp' '--enable-gd-native-ttf' '--with-bz2=/usr' '--with-ldap' '--with-mysql=mysqlnd' '--enable-mbstring=all' '--with-curl=/Applications/MAMP/Library' '--enable-sockets' '--enable-bcmath' '--with-imag-shared=/Applications/MAMP/Library/lib/imap-2007' '--enable-ldap' '--enable-kerberos' '--enable-calendar' '--with-pgsql-shared=/Applications/MAMP/Library' '--enable-xml' '--with-ldapdir=/Applications/MAMP/Library' '--with-gettext-shared=/Applications/MAMP/Library' '--with-xsl=/Applications/MAMP/Library' '--with-pdo-mysql=mysqlnd' '--with-pdo-pgsql-shared=/Applications/MAMP/Library' '--with-mcrypt-shared=/Applications/MAMP/Library' '--with-openssl' '--enable-zip' '--with-iconv=/Applications/MAMP/Library' '--enable-openssl' '--enable-intl' '--with-ldap-shared' '--with-icu-dir=/Applications/MAMP/Library' '--enable-wddx' '--with-ldap-dir=/Applications/MAMP/Library' '--with-readline' 'CFLAGS=-arch 'LDLAGS=-arch 'LIBS=-iresolv 'CXXFLAGS=-arch
Server API	Apache 2.0 Handler
Virtual Directory Support	disabled
Configuration File (php.ini) Path	/Applications/MAMP/bin/php/php7.0.0/conf
Loaded Configuration File	/Applications/MAMP/bin/php/php7.0.0/conf/php.ini
Scan this dir for additional .ini files	(none)
Additional .ini files parsed	(none)
PHP API	20151012
PHP Extension	20151012
Zend Extension	320151012
Zend Extension Build	API320151012.NTS
PHP Extension Build	API20151012.NTS
Debug Build	no
Thread Safety	disabled
Zend Signal Handling	disabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring

Figure 1-3

Informations concernant le serveur fournies par `phpinfo()`

Pour être conforme aux recommandations HTML 5, un document doit avoir la structure suivante (fichier `pagehtml.html`) :

```
<!DOCTYPE html>
<html lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<title>Titre de la page</title>
</head>
<body>
<h2>Bienvenue sur le site PHP 7 </h2>
</body>
</html>
```

Cette page primaire est écrite en HTML pur, et tous les visiteurs de votre site verront exactement le même contenu, quel que soit le moment de leur connexion. Le fichier peut avoir l'extension `.html` ou `.htm` car il ne contient que du code HTML, mais il pourrait tout aussi bien avoir une extension `.php` et avoir le même rendu dans un navigateur.

Vous pourriez lui apporter un brin de dynamisme en affichant la date du jour en tête de page à l'aide du code PHP suivant (fichier `codephp.php`) :

```
<!DOCTYPE html>
<html lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
```

```
<title>Une page PHP</title>
</head>
<body>
<?php
  echo "<h3> Aujourd'hui le ". date('d / M / Y H:m:s')."</h3><hr />";
  echo "<h2>Bienvenue sur le site PHP 7</h2>";
?>
</body>
</html>
```

Le code de votre nouvelle page contient les nouveaux éléments suivants, qui ne sont pas du HTML :

```
<?php
echo "<h3> Aujourd'hui le ". date('d / M / Y H:m:s')."</ h3><hr />";
echo "<h2>Bienvenue sur le site PHP 7</h2>";
?>
```

Les éléments `<?php` et `?>` marquent respectivement le début et la fin de tout script PHP, qu'il soit inclus dans du code HTML ou isolé dans un fichier ne contenant que du code PHP. Vous pouvez inclure autant de blocs de code PHP que vous le désirez dans un document HTML, à condition que chacun d'eux soit délimité par ces marqueurs.

Entre ces éléments figure le code PHP proprement dit :

```
echo "<h3> Aujourd'hui le ". date('d / M / Y H:m:s')."</ h3><hr />";
echo "<h2>Bienvenue sur le site PHP 7</h2>";
```

L'instruction `echo` permet d'écrire dans le document final le contenu qui la suit, que ce soit du texte ou le résultat retourné par une fonction, comme dans les deux lignes précédentes. Notez que les lignes de code PHP se terminent toujours par un point-virgule.

Si vous recopiez et exécutez ce fichier dans votre navigateur, vous obtenez le résultat illustré à la figure 1-5, qui donne un aperçu de ce qu'est une page dynamique élémentaire. Vous pourriez faire la même chose à l'aide d'un script JavaScript exécuté non pas sur le serveur mais par le navigateur du poste client. La différence est que la date et l'heure affichées ici sont celles du serveur et pas celle de votre ordinateur, comme le ferait JavaScript. L'un des avantages de PHP est cependant que vous n'avez pas à tenir compte des capacités du navigateur du visiteur.



Figure 1-4

Résultat de votre première page PHP

Examinez maintenant le code source du document tel qu'il a été reçu par le navigateur. Dans Opera, par exemple, allez dans le menu Développeur>Afficher la source. Le code suivant s'affiche :

```
<!DOCTYPE html>
<html lang="fr">
  <head>
    <meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
    <title>Une page PHP</title>
  </head>
  <body>
    <h3> Aujourd'hui le 22 / Jul / 2016 22:07:53</h3><hr />
    <h2>Bienvenue sur le site PHP 7</h2>
  </body>
</html>
```

Par rapport au code du fichier codephp.php, ce qui était contenu entre les éléments <?php et ?>, soit :

```
<?php
echo "<h3> Aujourd'hui le ". date('d / M / Y H:m:s ')."</h3><hr />";
echo "<h2>Bienvenue sur le site PHP 7</h2>";
?>
```

a été remplacé par :

```
<h3> Aujourd'hui le 22 / Jul / 2016 22:07:53</ h3><hr />
<h2>Bienvenue sur le site PHP 7</h2>
```

L'interpréteur PHP analyse le document dans son ensemble puis renvoie le code HTML tel quel, accompagné de l'évaluation des expressions contenues dans le code PHP. Cela fait d'ailleurs dire à certains que tout est expression dans PHP puisque tout le code peut être évalué comme une chaîne de caractères, un nombre ou une valeur booléenne.

Les parties de code contenues dans les guillemets sont renvoyées dans le flux du document HTML, et les balises qu'elles contiennent sont interprétées en tant que telles par le navigateur. C'est le cas de la deuxième ligne. La première ligne comporte une fonction PHP qui retourne la date du jour. Cette date est concaténée avec le texte qui l'entoure puis est retournée au navigateur.

Le cycle de vie d'une page PHP est le suivant :

- Envoi d'une requête HTTP par le navigateur client vers le serveur, du type *http://www.monserveur.com/codephp.php*.
- Interprétation par le serveur du code PHP contenu dans la page appelée.
- Envoi par le serveur d'un fichier dont le contenu est purement HTML.

Vous constatez ainsi que votre code PHP n'est jamais visible par les visiteurs de votre site.

Écriture du code PHP

Le code PHP est souvent incorporé dans du code HTML. Vous pouvez donc incorporer autant de scripts PHP indépendants que vous le souhaitez n'importe où dans du code HTML, du moment que ces parties sont délimitées par les balises ouvrantes et fermantes `<?php` et `"?"` (repères ❶, ❷, ❹, ❻ et ❼). Les formes `<?=>` et `?>` (repères ❸ et ❽) ou encore l'élément HTML `<script language="php">` (repère ❸), qui est rarement employé, sont désormais considérés comme obsolètes et déconseillés. Citée pour mémoire car certains l'utilisaient, cette dernière forme n'est même plus reconnue du tout par PHP 7 (voir figure 1-5), donc à proscrire.

Dans un fichier `.php`, vous pouvez à tout moment passer du code PHP au code HTML, et réciproquement. C'est ce qui donne sa grande souplesse d'utilisation à ce code.

Le listing suivant illustre cette particularité :

```
<!DOCTYPE html>
<?php ←❶
    $variable1=" PHP 7";
?>
<html lang="fr">
<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<?php ←❷
echo "<title>Une page pleine de scripts PHP</title>";
?>
</head>
<body>
<script language="php"> ←❸
    echo"<h1>BONJOUR A TOUS </h1>";
</script>
<?php ←❹
    echo "<h2> Titre écrit par PHP</h2>";
    $variable2=" MySQL";
?>
<p>Vous allez découvrir <?= $variable1 ?> ←❺</p>
<?php ←❻
    echo "<h2> Bonjour de $variable1</h2>";
?>
<p>Utilisation de variables PHP<br />Vous allez découvrir également
    <?php←❼
        echo $variable2
    ?>
</p>
<?= "<div><big>Bonjour de $variable2 </big></div>" ?> ←❽
</body>
</html>
```

Huit mini-scripts PHP sont placés aussi bien dans l'en-tête (entre `<head>` et `</head>`) que dans le corps (entre `<body>` et `</body>`) ou encore même en dehors du bloc délimité par les éléments `<html>` et `</html>` du document HTML.

Certains de ces scripts interviennent comme contenu d'un élément HTML avec une syntaxe particulière. Par exemple :

```
<?= $variable1 ?>
```

peut être utilisé pour des instructions courtes. Il est équivalent à :

```
<?php echo $variable1 ?>.
```

À partir de ce document, vous obtenez le résultat illustré à la figure 1-5.



Figure 1-5

Résultat des mini-scripts

Comme précédemment, la consultation du code source dans le navigateur montrerait que le résultat de chaque mini-script est purement HTML et qu'aucun code PHP ne subsiste.

Inclure des fichiers externes

Comme en JavaScript, il est possible d'écrire du code PHP ou HTML dans des fichiers séparés puis de les incorporer dans du code HTML ou d'autres scripts PHP en fonction des besoins. Cela peut constituer un début de modularisation du code, permettant d'écrire une seule fois certaines parties de code et de les réutiliser dans plusieurs pages différentes, avec économie de temps. Cette possibilité permet notamment de créer une bibliothèque de fonctions d'utilisation courante.

On donne généralement aux fichiers de code PHP l'extension `.inc` ou `.inc.php`, cette dernière ayant l'avantage de protéger les données confidentielles que peut contenir le code, comme les paramètres de connexion à la base de données (login et mot de passe). Le contenu du fichier est interprété par le serveur. Si le fichier ne contient que vos paramètres dans des variables, le serveur ne renvoie rien au poste client si quelqu'un tente de l'exécuter, alors qu'un navigateur affiche le contenu d'un fichier avec l'extension `.inc` seule.

Pour inclure le contenu d'un fichier externe dans du code PHP, vous disposez des fonctions recensées au tableau 1-2.

Tableau 1-2 – Fonctions d'inclusion de code externe

Fonction	Description
<code>include("nom_fichier.ext")</code>	Lors de son interprétation par le serveur, cette ligne est remplacée par tout le contenu du fichier précisé en paramètre, dont vous fournissez le nom et éventuellement l'adresse complète. En cas d'erreur, par exemple si le fichier n'est pas trouvé, <code>include()</code> ne génère qu'une alerte, et le script continue.
<code>require("nom_fichier.ext")</code>	A désormais un comportement identique à <code>include()</code> , à la différence près qu'en cas d'erreur, <code>require()</code> provoque une erreur fatale et met fin au script.
<code>include_once("nom_fichier.ext")</code> <code>require_once("nom_fichier.ext")</code>	Contrairement aux deux précédentes, ces fonctions ne sont pas exécutées plusieurs fois, même si elles figurent dans une boucle ou si elles ont déjà été exécutées une fois dans le code qui précède.

L'exemple suivant utilise les possibilités d'inclusion fournies par ces fonctions pour créer une page HTML à partir de quatre fichiers indépendants. Il s'agit d'un début de modularisation du code d'un site. Notre hypothèse est que chaque page du site a le même en-tête et le même pied de page et que chacune des pages ne diffère des autres que par son contenu.

L'exemple comprend les fichiers suivants :

- `tete.inc.php`. Contient le début du code HTML d'une page normale (`<html>`, `<head>`, `<body>`) et trois petits scripts PHP. Le dernier de ces scripts (repère ❶) affiche le bandeau commun à toutes les pages (repère ❷) ainsi que le nom du fichier exécuté et celui du fichier inclus (repère ❸).
- `corps.inc.php`. Ne contient que du code PHP affichant deux lignes de texte (repère ❹).
- `corps.html`. Ne contient que du code HTML affichant deux lignes de texte (repère ❺).
- `pied.inc.php`. Contient un script affichant un bandeau de pied de page et deux liens vers des sites dignes d'intérêt (repère ❻).
- `principal.php`. Script utilisant les quatre précédents à l'aide des fonctions `include()` (repère ❶), `include_once()` (repère ❷), `require()` (repère ❸) et `require_once()` (repère ❹). C'est le seul qui doit être appelé directement. Les autres fichiers n'étant que des composants, ils ne doivent normalement pas être utilisés seuls.

La figure 1-6 donne un aperçu du résultat obtenu.

Exemple 1-1. Inclusion de fichiers externes

Le fichier `tete.inc.php` :

```
<!DOCTYPE html>
<?php
    $variable1=" PHP 7";
?>
<html lang="fr">
```

```

<head>
<meta http-equiv="Content-Type" content="text/html; charset=UTF-8" />
<?php
echo "<title>Une page pleine d'inclusions $variable1</title>";
?>
</head>
<body>
<?php←❶
$variableext="Ce texte provient du fichier inclus";
echo "<div><h1 style=\"border-width:5;border-style:double;background-color:#ffcc99;\">
Bienvenue sur le site $variable1 </h1>";←❷
echo "<h3> $variableext</h3>";
echo "Nom du fichier ex&#233;cut&#233;: ", $_SERVER['PHP_SELF'],"&nbsp;&nbsp;&nbsp;"; ←❸
echo " Nom du fichier inclus : ", __FILE__ ,"</div> "; ←❹
?>

```

Le fichier corps.inc.php :

```

<?php
echo "<h1> Ceci est le corps du document </h1>"; ←❺
echo "<h2> Ceci est le corps du document </h2>";
?>

```

Le fichier corps.html :

```

<h1> Ceci est le corps du document : Avec PHP on progresse vite et avec MySQL le
➤ site devient vite tr&#232;s dynamique.....</h1>
<h2> On s'y met tout de suite!!!! </h2> ←❻

```

Le fichier pied.inc.php :

```

<hr />
<?php
echo "<div><h1 style=\"border-width:3;border-style:groove; background-color:
➤ #ffcc99;\"> Fin de la page PHP Liens utiles : <a href=\"php.net\">php.net</a>
➤ &nbsp;&nbsp;&nbsp; <a href=\"mysql.org\">mysql.org</a></h1>"; ←❼
echo "Nom du fichier ex&#233;cut&#233;: ", $_SERVER['PHP_SELF'],"&nbsp;&nbsp;&nbsp; &nbsp;&nbsp;";
echo "Nom du fichier inclus: ", __FILE__ ,"</div>";
?>
</body>
</html>

```

Le fichier principal.php :

```

<?php
include("tete.inc.php"); ←❶
echo "<hr />";
include_once("corps.inc.php"); ←❷
require("corps.html"); ←❸
require_once("pied.inc.php"); ←❹
?>

```



Figure 1-6

Un page composée de fichiers inclus

Ajout de commentaires

Il est toujours utile de commenter les scripts que vous écrivez. Lors de l'écriture, tout peut paraître évident, mais à la relecture, plusieurs mois plus tard, lorsqu'il s'agit d'effectuer des mises à jour, par exemple, autant éviter de perdre du temps à redécouvrir la logique adoptée auparavant.

Les commentaires ne sont pas pris en compte par l'analyseur PHP. S'ils alourdissent un peu le fichier PHP en termes d'octets sur le serveur, ils ne sont pas présents dans le code HTML renvoyé au navigateur client. Leur poids est donc sans importance pour la rapidité de transmission des pages.

PHP supporte les trois syntaxes de commentaires suivantes :

- commentaires sur une seule ligne introduits par les caractères `//` :

```
//Ceci est un commentaire court sur une ligne
```

- commentaires sur plusieurs lignes introduits par les caractères `/*` et fermés par les caractères `*/` :

```
/* Ceci est commentaire abondant
qui va occuper plusieurs lignes
et va expliquer le code qui suit..... */
```

- commentaires de type Unix, ne comportant qu'une seule ligne introduite par le caractère `#` :

```
##*****
# commentaires de type Unix
##*****
```