



Ressourcesinformatiques

Apprendre à développer un site web avec **PHP** et **MySQL**

Exercices pratiques et corrigés

4^e édition

Olivier ROLLET

Fichiers complémentaires
à télécharger





Les éléments à télécharger sont disponibles à l'adresse suivante :
<http://www.editions-eni.fr>
Saisissez la référence ENI de l'ouvrage **RI4PHMY** dans la zone de recherche et validez. Cliquez sur le titre du livre puis sur le bouton de téléchargement.

Chapitre 1 Introduction

- 1. Objectif de l'ouvrage 15
- 2. Fonctionnement d'un site web 16

Chapitre 2 Utilisation de WAMP

- 1. Introduction 19
- 2. Installation de WAMP 20
- 3. Prise en main de WAMP 20
- 4. Création d'une première page PHP 23

Chapitre 3 Les bases du langage PHP

- 1. Les balises 27
 - 1.1 Syntaxe de base 27
 - 1.2 Insertion des balises PHP dans du code XHTML 28
 - 1.3 Envoi des données au serveur web 30
 - 1.4 Insertion de code XHTML avec l'instruction echo 32
 - 1.5 Les commentaires 33
- 2. Les variables 33
 - 2.1 Affectation 33
 - 2.2 Les types de variables 34
 - 2.3 Particularité du type string 35

2.4	La concaténation	36
2.5	Le transtypage	38
2.6	Les variables dynamiques	40
3.	Les constantes	40

Chapitre 4

Les fonctions et structures de contrôle

1.	Les conditions	43
1.1	if	43
1.2	switch	48
2.	Les boucles	51
2.1	for	51
2.2	While	52
2.3	do while	54
2.4	foreach	54
3.	Les tableaux	55
3.1	Tableaux numériques	55
3.2	Tableaux associatifs	57
3.3	Constantes de type tableau	58
3.4	Parcours d'un tableau	59
3.4.1	La boucle for	59
3.4.2	La boucle foreach	60
3.4.3	La fonction print_r	63
3.5	Fonctions sur les tableaux	63
3.5.1	Longueur d'un tableau	63
3.5.2	Existence d'une valeur dans un tableau	64
3.5.3	Existence d'une clé dans un tableau	65
3.5.4	Tri d'un tableau	66
3.5.5	Recherche dans un tableau	70
3.5.6	Découper une chaîne dans un tableau	71
3.5.7	Regrouper les valeurs d'un tableau dans une chaîne	72

3.5.8	Découper une chaîne en morceaux de longueur fixe . . .	73
3.5.9	Ajouter des éléments à la fin du tableau	74
3.5.10	Suppression d'un élément à la fin du tableau	75
3.5.11	Sélection aléatoire d'un élément du tableau	76
3.6	Tableaux à plusieurs dimensions	77
3.7	Exercices sur les tableaux	79
3.7.1	Énoncés	79
3.7.2	Corrigés	81
4.	Traitement de chaînes de caractères	84
4.1	Les fonctions de manipulation de chaîne	84
4.1.1	strlen()	84
4.1.2	substr()	85
4.1.3	strstr()	86
4.1.4	str_replace()	87
4.1.5	trim()	88
4.1.6	strtolower()	88
4.1.7	strtoupper()	89
4.1.8	strpos()	90
4.1.9	str_word_count()	91
4.1.10	str_pad()	92
4.2	Les expressions régulières	93
4.2.1	Insensibilité à la casse	95
4.2.2	Recherche d'un mot et non d'une chaîne	95
4.2.3	Recherches de chaînes avec OU	97
4.2.4	Début de chaîne	98
4.2.5	Fin de chaîne	98
4.2.6	Un caractère dans une classe	99
4.2.7	Une plage de caractères dans une classe	100
4.2.8	La non-présence d'une plage de caractères dans une classe	101
4.2.9	Les quantificateurs	102
4.2.10	Les intervalles de reconnaissance	103

5.	Les opérateurs	106
5.1	Les opérateurs de chaîne.	106
5.1.1	La concaténation	106
5.1.2	L'assignation	106
5.2	Les opérateurs arithmétiques.	107
5.2.1	L'addition	107
5.2.2	La soustraction	108
5.2.3	La multiplication	108
5.2.4	La division	109
5.2.5	Le modulo	109
5.2.6	L'incrémentation.	110
5.2.7	La décrémentation	111
5.2.8	L'exponentiation.	112
5.3	Les opérateurs de comparaison	113
5.3.1	L'égalité	113
5.3.2	La différence	114
5.3.3	La comparaison.	115
5.3.4	L'opérateur à trois issues	116
5.4	L'opérateur ternaire	116
5.5	Les opérateurs logiques.	118
5.5.1	ET	118
5.5.2	OU.	118
6.	Les fonctions.	121
6.1	Création	121
6.2	Les exceptions.	124
6.3	Retour du type des déclarations	125
6.4	Portée des variables.	128
6.5	Les variables globales	129
6.6	Les variables statiques	130
6.7	Fonctions utiles.	131
6.8	Passage par référence	134
6.9	Fonctions sur la gestion de fonction	135
6.10	Fonctions variables via l'opérateur	137

6.11	Décompression des arguments via l'opérateur	138
6.12	Fonction anonyme	139
6.13	Fonctions dynamiques interdites	141
6.14	Fonction générateur	142
6.15	Récurtivité.	143
6.16	Fonctions prédéfinies dans PHP.	145
6.16.1	Générer un nombre aléatoire	145
6.16.2	Arrondir un nombre décimal	146
6.16.3	Récupérer la valeur absolue d'un nombre	147
6.16.4	Créer un identifiant unique	148
6.16.5	Afficher les informations sur PHP.	148
6.16.6	Envoyer un e-mail.	150
6.17	Stocker une fonction dans une variable	151
6.18	Gérer la mémoire tampon	152
6.19	Exercices sur les fonctions	153
6.19.1	Énoncés	153
6.19.2	Solutions	154
7.	Les dates	157
8.	Les fichiers.	166
8.1	Introduction	166
8.2	Lecture rapide	167
8.3	Écriture rapide.	168
8.4	Ouverture et fermeture d'un fichier	169
8.5	Lecture et écriture.	170
8.6	Accès concurrent.	177
8.7	Manipulation de fichiers	178
8.8	Manipulation de répertoires.	181
9.	Les includes	188
10.	Exercices sur les fichiers	190
10.1	Énoncés	190
10.2	Corrigés	190

Chapitre 5**Transmettre des données d'une page à l'autre**

1.	Les variables superglobales	193
1.1	\$GLOBALS	193
1.2	\$_SERVER	194
1.3	\$_ENV	198
1.4	\$_SESSION	200
1.5	Les options de Session	202
1.6	\$_COOKIE	202
1.7	\$_FILES	205
2.	La méthode GET	208
2.1	Utilisation de la méthode GET	208
2.2	Test de la présence de la variable dans l'URL	210
2.3	Test de la valeur de la variable dans l'URL	212
2.4	Informations complémentaires	214
2.4.1	Paramètres avec le même nom	214
2.4.2	Paramètres de type tableau	214
2.4.3	Paramètres avec des caractères spéciaux	215
3.	La méthode POST	217
3.1	Utilisation de la méthode POST	217
3.2	Les différents éléments du formulaire	219
3.2.1	Zone de type texte	220
3.2.2	Zone de type mot de passe	221
3.2.3	Grande zone de texte	222
3.2.4	Liste déroulante à choix unique	223
3.2.5	Liste déroulante à choix multiple	224
3.2.6	Cases à cocher	226
3.2.7	Boutons radio	228
3.2.8	Champs cachés	229
3.2.9	Bouton de type submit	230
3.2.10	Bouton de type reset	233
3.2.11	Formulaire complet	234

- 4. La méthode `$_REQUEST` 236
- 5. Extraction de données 239
- 6. Zones portant le même nom 239
- 7. Plusieurs formulaires dans la même page 240
- 8. Contrôle des données et redirection de pages 240
 - 8.1 Introduction 240
 - 8.2 Données obligatoires 241
 - 8.3 Les filtres 241
 - 8.4 Suppression des espaces indésirables 242
 - 8.5 Longueur maximale 243
 - 8.6 Caractères autorisés 243
 - 8.7 Magic quotes 245
 - 8.8 Redirection de page 246
- 9. Exercices 248
 - 9.1 Énoncés 248
 - 9.2 Solutions 249

Chapitre 6
Les effets spéciaux sur une image

- 1. La librairie GD 255
- 2. Création d'une image 255
 - 2.1 Le header 255
 - 2.2 Création d'une image vide 256
 - 2.3 Création et affichage d'une image complète 256
 - 2.4 Résolution de l'image 258
- 3. Texte et couleur 259
 - 3.1 La couleur 259
 - 3.2 Le texte 260
 - 3.3 La transparence 261
- 4. Redimensionnement d'une image 261

5.	Superposition d'images	262
6.	Les formes	266
7.	Différents exemples	268
7.1	Exemple 1	268
7.2	Exemple 2	270

Chapitre 7

Base de données MySQL

1.	Présentation	271
1.1	Introduction	271
1.2	Structure	272
2.	phpMyAdmin	273
3.	Le langage SQL	285
3.1	Présentation	285
3.2	Lire des données	286
3.3	Écrire des données	288
3.4	Filtrer des données	292
3.5	Les alias	294
3.6	Tri des données	295
3.7	Supprimer des données	296
3.8	Modifier des données	297
3.9	Les jointures	299
3.10	Le regroupement	303
4.	SQL avancé	306
4.1	Les fonctions et instructions SQL	306
4.1.1	Limiter des données	306
4.1.2	Valeurs distinctes	307
4.1.3	Convertir en majuscules	308
4.1.4	Convertir en minuscules	309
4.1.5	Arrondir un nombre décimal	310
4.1.6	Valeur absolue d'un nombre décimal	310

4.1.7	Nombre aléatoire	310
4.1.8	Longueur d'un champ	311
4.1.9	Supprimer les espaces d'un champ	311
4.1.10	Extraire une sous-chaîne d'un champ	312
4.1.11	Concaténer plusieurs champs	312
4.1.12	Position d'une chaîne de caractères dans un champ	313
4.1.13	Ajout d'une séquence de caractères	314
4.1.14	Remplacement d'une chaîne de caractères	314
4.1.15	Tester la valeur d'un champ	315
4.1.16	Affichage de la date courante	316
4.1.17	Extraire la date d'un champ date et heure	317
4.1.18	Différence entre deux dates	317
4.1.19	Ajout d'un intervalle de temps à une date	318
4.1.20	Ajout d'un intervalle de temps à une heure	319
4.1.21	Soustraction d'un intervalle de temps à une date	320
4.1.22	Soustraction d'un intervalle de temps à une heure	320
4.1.23	Unir deux requêtes	321
4.2	Les fonctions de chiffrement et de compression	322
4.3	La commande SHOW	324
4.4	La commande SHOW PROCESSLIST	325
4.5	Les expressions rationnelles	326
4.6	Les sous-requêtes	327
4.7	Les procédures stockées et les fonctions	331
4.8	Les autres objets de MySQL	338
4.8.1	Les tables	338
4.8.2	Les index	338
4.8.3	Les vues	340
4.8.4	Les déclencheurs	340
5.	Exercices SQL	342
5.1	Énoncés	344
5.2	Corrigés	347

6.	L'accès aux bases de données avec PHP	349
6.1	Introduction	349
6.2	Connexion.	350
6.3	Déconnexion.	351
6.4	Requêtes non préparées	352
6.4.1	Lire des données	352
6.4.2	Écrire des données.	358
6.4.3	Supprimer des données	359
6.4.4	Mettre à jour des données	361
6.5	Requêtes préparées	362
6.5.1	Introduction	362
6.5.2	Lire des données	363
6.5.3	Écrire des données.	366
6.5.4	Modifier des données	367
6.5.5	Supprimer des données	368
6.5.6	Stocker un résultat	370
6.5.7	Afficher les erreurs d'une requête préparée.	372
7.	PDO.	373
7.1	Introduction	373
7.2	Connexion.	374
7.3	Requêtes non préparées	377
7.3.1	Lire des données	377
7.3.2	Écrire des données.	379
7.3.3	Supprimer des données	381
7.3.4	Mettre à jour des données	382
7.4	Requêtes préparées	383
7.4.1	Lire des données	383
7.4.2	Écrire des données.	385
7.4.3	Supprimer des données	387
7.4.4	Modifier des données	388
7.4.5	Appeler une procédure stockée	389

8. Exercices	391
8.1 Énoncés	391
8.2 Corrigés	393

Chapitre 8 L'objet

1. Introduction	403
2. Les classes	404
2.1 Introduction	404
2.2 L'encapsulation	405
2.3 Visibilité des attributs et des méthodes	405
2.4 Ajout d'une méthode dans une classe	406
2.5 Utilisation d'une classe	407
2.6 Mettre à jour et lire les attributs de l'instance	408
2.7 Passage en paramètre de type objet	411
2.8 Le constructeur	414
2.9 Le destructeur	416
2.10 Exercice	417
2.11 Les constantes de classe	419
2.12 Les attributs et méthodes statiques	421
2.12.1 Méthode statique	421
2.12.2 Attribut statique	424
2.13 Les exceptions	426
3. L'héritage	428
3.1 Introduction	428
3.2 Protected	432
3.3 Substitution	434
3.4 Héritage en cascade	436
4. Les classes abstraites	437
5. Substitution de méthodes abstraites	440
6. Les interfaces	441

7.	Les classes finales	442
8.	Les méthodes magiques	444
9.	Les classes anonymes	452
10.	Les traits	453
11.	Uniform Variable Syntax	454
12.	Les espaces de noms	455
13.	Autoload	460
14.	Exercices	460
	14.1 Énoncés	460
	14.2 Corrigés	466

Chapitre 9

Configuration

1.	Fichier de configuration PHP : php.ini	487
2.	Fichier de configuration MySQL : my.ini	489
3.	Fichier de configuration Apache : httpd.conf	490

Chapitre 10

Sécurité

1.	Faibles XSS	493
	1.1 XSS non permanent	493
	1.2 XSS permanent	495
	1.3 Page d'erreur	495
2.	Droits sur la base de données	496
3.	Injection SQL (addslashes)	498
4.	Vérification de la session	500

5.	Performances.....	501
5.1	PHP 7.....	501
5.2	Utilisation d'AST.....	502
5.3	Optimiser les performances en PHP.....	503
5.4	Optimiser les performances en MySQL.....	504

Chapitre 11

Cas pratiques et corrigés

1.	Créer un blog (procédural).....	505
2.	Créer un blog (objet).....	510
3.	Créer une newsletter.....	518
4.	Créer un flux RSS.....	519
5.	Gestion d'un parc informatique en MVC.....	520
6.	Créer un site de gestion d'élèves.....	531
7.	Génération d'un RIB en fichier PDF.....	551
8.	Paiement par PayPal.....	557

Index.....	571
------------	-----



Chapitre 4

Les fonctions et structures de contrôle

1. Les conditions

1.1 if

Les instructions de type condition permettent d'exécuter du code si une condition est vraie.

Par exemple, si le prénom est égal à Robert, afficher "Bienvenue" :

```
<?php
$prenom = 'Robert'; //déclaration de la variable $prenom

if ($prenom == 'Robert') //test de la variable $prenom
{
    echo 'Bienvenue';
}
?>
```

Ci-dessus, la valeur Robert est affectée à la variable \$prenom puis cette variable est testée à l'aide de l'instruction **if**.

La syntaxe est donc :

```
if (condition) { instruction }
```

Notez que :

- "est égal à" se note `==`.
- "est différent de" se note `!=`.
- "est inférieur à" se note `<`.
- "est supérieur à" se note `>`.
- "est inférieur ou égal à" se note `<=`.
- "est supérieur ou égal à" se note `>=`.

L'instruction **sinon** se note **else**.

Par exemple, si le prénom est égal à Robert, afficher "Bienvenue", sinon "A bientôt" :

```
<?php
    $prenom = 'Toto'; //déclaration de la variable $prenom

    if ($prenom == 'Robert') //test de la variable $prenom
    {
        echo 'Bienvenue';
    }
    else
    {
        echo 'A bientôt';
    }
?>
```

Affiche :

A bientôt

En effet, le code teste si la variable `$prenom` est égale à Robert, puis comme ce n'est pas le cas, le code passe dans le **else** (sinon) et exécute `echo "A bientôt";`.

Enfin la dernière instruction pour les conditions est **else if** appelée **sinon si**.

Cela permet de tester d'autres conditions non testées par le **if**.

Par exemple, si le prénom est égal à Robert, afficher "Bienvenue", sinon s'il est égal à Toto, afficher "Bonjour", sinon "A bientôt" :

```
<?php
$prenom = 'Toto'; //déclaration de la variable $prenom

if ($prenom == 'Robert') //test de la variable $prenom
{
    echo 'Bienvenue';
}
else if ($prenom == 'Toto') //test de la variable $prenom
{
    echo 'Bonjour';
}
else
{
    echo 'A bientôt';
}
?>
```

Affiche :

Bonjour

Le code teste si \$prenom est égal à Robert puis si \$prenom est égal à Toto. Et comme \$prenom est effectivement égal à Toto, le code exécute echo "Bonjour";.

Il est possible d'ajouter autant d'instructions **else if** que vous voulez.

Si la condition (`$prenom == "Robert"`) est vérifiée, les autres conditions dans les **else if** ne sont même pas vérifiées, donc si vous passez dans le **if**, vous êtes sûr de ne jamais passer dans le ou les **else if** ainsi que dans le **else**.

À l'inverse, si vous écrivez :

```
<?php
$prenom = 'Robert'; //déclaration de la variable $prenom

if ($prenom == 'Robert') //test de la variable $prenom
```

```
{
    echo 'Bienvenue';
}

if ($prenom == 'Toto') //test de la variable $prenom
{
    echo 'Bonjour';
}
else
{
    echo 'A bientôt';
}
?>
```

Le code teste si \$prenom est égal à Robert puis si \$prenom est égal à Toto et comme \$prenom est initialisé avec la valeur Robert, ce code affiche "Bienvenue" et "A bientôt".

Vous pouvez aussi imbriquer les **if** les uns à l'intérieur des autres autant de fois que vous le souhaitez.

Par exemple :

```
<?php

$age = 30; //déclaration de la variable $age

if ($age > 20) //test de la variable $age
{
    if ($age == 30) //test de la variable $age
    {
        echo 'Bienvenue';
    }
    else {
        echo 'A bientôt';
    }
}
?>
```

Affiche :

Bienvenue

Le programme teste si la variable \$age est supérieure à 20 puis comme c'est le cas, le programme teste ensuite si \$age est égal à 30 et exécute echo "Bienvenue";.

Enfin, vous pouvez intercaler du code HTML entre les conditions en PHP.

Par exemple :

```
<?php
$prenom = 'Robert'; //déclaration de la variable $prenom

if ($prenom == 'Robert') //test de la variable $prenom
{
?>
    Bonjour <!--Code HTML-->
<?php
}
else
{
?>
    A bientôt <!--Code HTML-->
<?php
}
?>
```

est équivalent à :

```
<?php
$prenom = 'Robert'; //déclaration de la variable $prenom

if ($prenom == 'Robert') //test de la variable $prenom
{
    echo 'Bonjour';
}
else
{
    echo 'A bientôt';
}
?>
```

1.2 switch

L'instruction **switch** est équivalente au **if** mais elle est utilisée par le développeur pour plus de clarté dans le code. Elle est équivalente au **if** si vous utilisez un **break** pour sortir du **switch**. Sinon toutes les instructions qui suivent le **case** dans lequel vous êtes entré seront exécutées.

La syntaxe est :

```
switch (condition) {  
    case expression: instruction  
    case expression: instruction  
    ...  
}
```

```
<?php  
  
$prenom = 'Robert'; //déclaration de la variable $prenom  
  
switch ($prenom) //test de la variable $prenom  
{  
    case 'Robert':  
        echo 'Bonjour';  
        break;  
    case 'Jean':  
        echo 'A bientôt';  
        break;  
}  
?>
```

L'instruction **break** provoque la sortie du **switch** et donc si **\$prenom** est égal à "Robert", le code exécutera `echo "Bonjour";` puis **break** et sortira du **switch** sans même tester "Jean".

Cela équivaut à utiliser l'instruction **else if** à la place de **if** :

```
<?php  
  
$prenom = 'Robert'; //déclaration de la variable $prenom  
  
if ($prenom == 'Robert') //test de la variable $prenom  
{  
    echo 'Bonjour';  
}
```