

Préparation à la Certification **ZCPE** **Zend Certified PHP Engineer** **PHP**

Préface de **Laurent DAUÇA**
Responsable Grands Comptes et Partenariat Zend

45 Travaux pratiques
156 Questions-réponses

OFFERT :
UN EXAMEN BLANC en ligne
avec réponses commentées et détaillées



Téléchargement
www.editions-eni.fr



Julien CHARPENTIER
Cyrille GRANDVAL

Les éléments à télécharger sont disponibles à l'adresse suivante :

<http://www.editions-eni.fr>

Saisissez la référence ENI du livre CEZCPE dans la zone de recherche et validez.

Cliquez sur le titre du livre puis sur le bouton de téléchargement.

Préface

Descriptif

Chapitre 1

Introduction

A. La certification	20
1. Pourquoi devenir certifié ?	20
2. L'historique	21
a. Les certifications	21
b. La société Zend	21
3. Environnement de la certification	21
a. Environnement	21
b. Indépendance	21
4. Les thèmes abordés	22
B. Modalités de l'examen	22
1. Comment s'inscrire et où le passer ?	22
a. Achat du voucher	22
b. Inscription à l'examen	22
c. Passage de l'examen	23
2. L'examen	23
a. Approche des questions	23
b. Marque et revue des questions	23
c. Trois types de questions	24
d. Importance des questions	24
e. Les réponses	24
3. Le résultat	25
C. L'ouvrage	25
1. Les chapitres	25
2. La configuration nécessaire	25
3. La définition des fonctions	26

Chapitre 2**Bases PHP**

A. Configuration	28
1. Fichier de configuration php.ini	28
a. Principe	28
b. Directive d'accès global	28
c. Directive d'échappement automatique	28
d. Gestion des erreurs	29
e. Modules/extensions	32
2. Fichier de configuration user.ini	33
B. Performances	34
1. Garbage collection (ramasse-miettes)	34
2. Cache bytecode	36
3. Extension OPcache	36
C. Syntaxe	37
1. Balises	37
2. Ponctuation	38
3. Commentaires	39
4. Blocs de code	39
D. Types de données	40
1. Définition	40
2. Booléen	40
3. Nombre entier	40
4. Nombre flottant	41
5. Chaîne de caractères	41
6. Tableau	42
7. Objet	42
8. Ressource	42
9. Nul	42
E. Transtypage	42
1. Définition	42
2. Vers une chaîne de caractères	43
3. Vers un nombre	43
4. Vers un booléen	44
5. Forcer une conversion	44
F. Constantes et variables	45
1. Variables	45
a. Règles de nommage	45
b. Variables dynamiques	45

c. Portée	45
d. Typage	46
e. Existence d'une variable	46
f. Destruction d'une variable	47
2. Constantes	47
a. Règles de nommage et définition	47
b. Portée	48
c. Typage	48
G. Espaces de noms	48
1. Définition	48
2. Déclaration et accès	49
a. Déclaration d'un espace de noms	49
b. L'espace de noms global	50
c. Accès à un espace de noms	51
H. Opérateurs	54
1. Opérateurs arithmétiques	54
2. Opérateurs d'incrément/décément	54
3. Opérateurs sur les bits	55
4. Opérateurs d'affectation	55
5. Opérateurs de comparaison	56
6. Opérateurs de tableaux	57
7. Opérateurs logiques	57
8. Opérateur d'exécution	58
9. Opérateurs combinés (notation courte)	58
10. Priorités entre opérateurs	59
I. Structures de contrôle	60
1. Conditions	60
2. Boucles itératives	63
a. Structure et définition des boucles	63
b. Mots-clés break et continue	65
J. Structures et fonctions du langage	66
1. Définition	66
2. Exemples de structures du langage	67
a. Structures de sortie	67
b. Structures d'affichage	68
c. Structures d'évaluation	69
d. Autres types de structures	70
K. Validation des acquis : questions/réponses	71

L. Travaux pratiques	73
1. Récupération de l'architecture	74
2. Configuration de PHP	74
a. Création et configuration du fichier .htaccess	74
3. Constantes de l'application	77
a. Création des constantes nécessaires à l'application	77
4. Routage des pages	78
a. Mise en place d'un système de routage	78
b. Page par défaut	80
5. Création des pages de l'application	82
a. Table de multiplication	82
b. Table de multiplication en affichage binaire	87
c. Plateau d'échec	89
d. Opération mathématique	95

Chapitre 3

Formats et types de données

A. XML	100
1. Définition	100
2. Extension	100
B. Manipulation XML	100
1. XML Parser	100
a. Définition	100
b. Fonctions	100
2. SimpleXML	104
a. Définition	104
b. Fonctions	104
c. Classes	105
3. DOM	108
a. Définition	108
b. Constantes	108
c. Interaction avec SimpleXml	108
C. JSON	109
1. Définition	109
2. Constantes	110
3. Fonctions	110
4. Interface	114

D. Principes de base des services web	115
1. SOAP	115
a. Définition	115
b. Installation	115
c. Configuration	116
d. Constantes	116
e. Fonctions	117
f. Classes	118
2. REST	119
a. Définition	119
b. Principes	119
c. Vocabulaire	119
d. Types de données	120
e. Verbes HTTP	120
f. Changement de contexte	121
g. Code HTTP	121
h. Consommation d'un service REST en PHP	122
E. Date et heure	122
1. Définition	122
2. Configuration	122
3. Classe DateTime	122
a. Constantes	122
b. Méthodes	123
F. Validation des acquis : questions/réponses	126
G. Travaux pratiques	129
1. Récupération de l'architecture	129
2. Présentation de l'application	129
a. Description globale	129
b. Description des fichiers non modifiables	129
c. Description des fichiers à modifier	129
3. Réalisation de l'application	130
a. Liste des utilisateurs affichée en brut	130
b. Liste des utilisateurs au format XML	131
c. Liste des prénoms utilisateurs à l'aide de XML Parser	132
d. Liste des prénoms utilisateurs à l'aide de XPath	134
e. Affichage de la date de naissance du second utilisateur à l'aide de DOM	135
f. Liste des utilisateurs au format JSON	137

g. Travail sur les dates	138
h. Gestion d'un serveur SOAP	140
i. Gestion d'un serveur REST	143

Chapitre 4

Chaînes de caractères et motifs

A. Définition	150
1. Apostrophes	150
2. Guillemets	151
3. HEREDOC et NOWDOC	151
4. Accès à un caractère	152
5. Valeur ASCII	153
B. Comparaison	153
1. Égalité	153
2. Différence/similitude	154
C. Comptage	155
1. Calcul de la taille	155
2. Nombre d'occurrences d'un segment	156
3. Nombre de mots	157
D. Formatage	158
1. Masque des fonctions de formatage	158
2. Fonctions de formatage	159
E. Expressions régulières	161
1. Définition	161
2. Délimiteurs et modificateurs	161
a. Délimiteurs	161
b. Modificateurs	162
3. Échappement	162
4. Métacaractères, classes et codes spéciaux	163
a. Métacaractères	163
b. Classes	164
c. Codes spéciaux	165
5. Répétitions	166
6. Sous-masques et captures	167
F. Recherche de caractères et positions	167
1. Recherche d'une position	167
2. Recherche de caractères non souhaités	169
3. Recherche à l'aide d'expressions régulières	169

G. Extraction	172
1. Création de chaînes	172
2. Création de tableaux	174
a. Extraction de chaînes à l'aide de fonctions standards	174
b. Extraction de chaînes à l'aide d'expressions régulières	176
H. Phonétique	178
I. Remplacement	179
1. Remplacement à l'aide de fonctions standards	179
2. Remplacement à l'aide d'expressions régulières	181
J. Encodage	183
K. Validation des acquis : questions/réponses	184
L. Travaux pratiques	187
1. Récupération de l'architecture	187
2. Configurer PHP	187
a. Installation du module mbstring	187
3. Page de rendu de l'analyseur	189
a. Affichage du titre	189
b. Extrait	191
c. Phrase aléatoire	193
d. Statistiques	195
e. Correction	198

Chapitre 5

Tableaux

A. Types de tableaux	202
1. Tableaux à indexation numérique	203
2. Tableaux associatifs (à clés littérales)	204
3. Transtypage de clé	205
4. Tableaux multidimensionnels	206
B. Informations	207
1. Taille	207
2. Accès à un élément	208
a. Accès direct	208
b. Accès indirect	209
3. Déréférencement littéral	210

C. Remplissage	210
1. Remplissage manuel	210
2. Fonctions de remplissage d'un tableau	211
D. Gestion de la pile	214
1. Ajout d'un élément	214
2. Retrait d'un élément	217
E. Itérations	219
1. Structure foreach	219
a. Utilisation par copie	219
b. Utilisation par référence	221
c. Utilisation avec list	222
2. Autre structure	223
3. Fonctions permettant l'itération d'un tableau	223
F. Recherche	224
1. Recherche d'une valeur	224
2. Recherche d'une clé	227
G. Extraction	227
H. Tri	229
1. Tri sur les valeurs	229
2. Tri sur les clés	232
I. Comparaison	233
1. Différence	233
a. Fonction principale	233
b. Fonctions relatives	234
2. Intersection	235
a. Fonction principale	235
b. Fonctions relatives	236
J. Fusion et fission	236
1. Fusion	236
2. Fission	241
K. SPL, objets comme tableaux	242
1. Constructeur	242
2. Méthodes	244
L. Validation des acquis : questions/réponses	244

M. Travaux pratiques	249
1. Récupération de l'architecture	249
2. Présentation de l'application	250
a. Description globale	250
b. Description des fichiers non modifiables	250
c. Description des fichiers à modifier	250
3. Réalisation de l'application	250
a. Fonctionnement nominal	250
b. Ajout d'une table	251
c. Existence d'une table	253
d. Récupération d'une table	254
e. Nombre d'éléments d'une table	256
f. Sélection de tous les champs d'une table	257
g. Sélection ordonnée d'une table sur un champ	259
h. Sélection d'un champ d'une table	262
i. Sélection des enregistrements d'une table répondant à des critères	264
j. Sélection limitée des enregistrements d'une table	268

Chapitre 6

Gestion des flux (entrées/sorties)

A. Configuration	272
B. Flux	273
1. Gestionnaires (wrappers)	274
a. Gestionnaires prédéfinis	274
b. Gestionnaires personnalisés	275
c. Liste des gestionnaires de flux disponibles	277
2. Socket	278
a. Ouverture d'une socket Internet ou Unix	278
b. Connaître l'état d'une socket	279
c. Fermeture d'une socket	280
3. Contextes	280
a. Création d'un contexte	281
b. Création d'un contexte vide	282
c. Mise à jour des options d'un contexte	282
d. Mise à jour des paramètres d'un contexte	282
e. Spécifier les options de contexte par défaut de tous les flux	283

4. Filtres	283
a. Ajout d'un filtre au début de la pile.	283
b. Ajout d'un filtre à la fin de la pile.	284
c. Création d'un filtre personnalisé	284
d. Utilisation de filtres avec les fonctions d'accès rapide	285
C. Manipulation de fichiers	286
1. Fonctions d'accès rapide.	286
a. Lecture	286
b. Écriture	288
c. Taille d'un fichier	289
2. À travers une ressource.	290
a. Mode d'ouverture	290
b. Création et fermeture.	291
c. Fin de fichier.	291
d. Lecture	292
e. Écriture	295
3. Opérations sur les fichiers et dossiers	297
a. Vérification de l'existence d'un fichier/dossier	297
b. Vérification de l'accessibilité en lecture	297
c. Opérations sur les fichiers	298
d. Opérations sur les dossiers	300
D. Validation des acquis : questions/réponses	301
E. Travaux pratiques	304
1. Récupération de l'architecture.	304
2. Configurer PHP	304
a. Présence de mbstring.	304
b. Manipulation de fichiers - Accès aux URL	305
3. Création du dossier de log	306
4. Page de rendu de l'analyseur.	307
a. Enregistrement des données saisies dans un fichier.	307
b. Récupération du contenu d'une URL	310
5. Page de dernière analyse	312
a. Récupération du contenu de la dernière analyse	312
b. Ajout d'un filtre au contenu.	314

Chapitre 7	Fonctions
A. Bases	318
1. Propriétés	318
2. Déclaration	318
B. Arguments	319
1. Utilisation	319
2. Fonctions	320
a. Fonction func_num_args	320
b. Fonction func_get_args	321
c. Fonction func_get_arg	321
C. Valeur de retour	322
D. Portée des variables	323
E. Fonctions variables	324
F. Fonctions anonymes	324
1. Définition	324
2. Héritage du contexte parent	325
3. Utilisation du contexte objet	326
G. Fonctions internes utiles	327
1. Fonction function_exists	327
2. Fonction is_callable	327
3. Fonction call_user_func	328
4. Fonction call_user_func_array	331
H. Validation des acquis : questions/réponses	331
I. Travaux pratiques	334
1. Récupération de l'architecture	334
2. Présentation de l'application	334
a. Description globale	334
b. Description des fichiers non modifiables	334
c. Description des fichiers à modifier	335
3. Réalisation de l'application	335
a. Création de la fonction utilisateur	335
b. Appel d'une fonction	336
c. Appel d'une méthode statique	338
d. Appel d'une méthode non statique	340
e. Prise en charge d'un argument	341
f. Prise en charge de plusieurs arguments	343
g. Prise en charge de fonctions variables et anonymes	346

Chapitre 8**Programmation orientée objet**

A. Introduction	352
B. Classes	352
1. Syntaxe	352
a. Implémentation d'une classe	352
b. Instanciation d'un objet	353
2. Attributs	353
3. Méthodes	354
4. Visibilité	354
5. Méthodes et attributs statiques	355
6. Constantes	357
7. Constructeur et destructeur	357
a. Constructeur	357
b. Destructeur	358
8. Méthodes magiques (__*)	359
a. Définition	359
b. Nouvelles méthodes magiques	360
C. Héritage	362
1. Principe	362
a. Définition	362
b. Héritage et visibilité	363
c. Héritage et constante	363
2. Polymorphisme	364
3. Le mot-clé final	365
D. Abstraction	366
1. Principe	366
2. Syntaxe	366
E. Interfaces	368
1. Principe	368
2. Syntaxe	368
a. Déclaration d'une interface	368
b. Implémentation d'une interface	369
3. Constantes	369
4. Méthodes	370
5. Héritage	371

F. Traits	372
1. Principe	372
2. Syntaxe	372
a. Déclaration d'un trait	372
b. Utilisation d'un trait	372
3. Attributs et méthodes	373
4. Surcharge et précédence	375
5. Visibilité et alias de méthodes	376
6. Déclaration de traits multiples	377
7. Priorité	378
8. Traits composites	379
9. Périmètre et champ d'action	380
G. Exceptions	381
1. Principe	381
2. Classe Exception	382
3. Héritage	383
4. Interception et levée	383
H. Autochargement	387
I. Objets	388
1. Référence	388
2. Typage d'argument	388
3. Comparaison	389
4. Clonage	391
5. Transtypage en chaîne de caractères	392
6. Itération	393
7. Sérialisation	394
8. Méthodes de « surcharge » magiques	396
a. Surcharge magique des attributs (<code>__set</code> et <code>__get</code>)	396
b. Surcharge magique des attributs (<code>__isset</code> et <code>__unset</code>)	398
c. Surcharge magique de méthodes (<code>__call</code> et <code>__callStatic</code>)	400
J. Liaison statique tardive (Late Static Bindings)	401
K. Réflexion	404
1. Principe	404
2. Application	404
3. Exemples d'utilisation	405
L. SPL	410
1. Interfaces	410
a. Iterator	410

b. ArrayAccess	413
c. Countable	415
d. RecursivelyIterator	416
e. Autres interfaces	418
2. Classes	418
a. RecursivelyIteratorIterator	418
b. ArrayIterator	419
c. ArrayObject	420
d. Autres classes	421
3. Autochargement	421
4. Fonctions utiles	422
a. iterator_apply	422
b. iterator_to_array	423
c. Autres fonctions	423
M. Motifs de conception	424
1. Singleton	424
2. Modèle, Vue, Contrôleur (Model, View, Controller)	425
3. Fabrique (Factory)	427
4. Itérateur (Iterator)	427
5. Délégation (Proxy)	428
6. Sujet observateur (Subject Observer)	430
7. Enregistrement actif (Active Record)	433
N. Validation des acquis : questions/réponses	434
O. Travaux pratiques	440
1. Récupération de l'architecture	441
2. Présentation de l'application	441
a. Description globale	441
b. Description de l'architecture	441
c. Description de la nomenclature des fichiers	442
3. Réalisation du framework MVC	442
a. Exception de base de l'application	443
b. Classe représentant la requête HTTP	443
c. Classe représentant la réponse HTTP	448
d. Classe représentant le routeur	451
e. Classe représentant la vue	453
f. Classe représentant le contrôleur d'action	457
g. Classe représentant le contrôleur frontal	459
h. Affichage de l'application	460

Chapitre 9**Bases de données**

A. Rappels	466
B. Clés	467
1. Clé primaire	467
2. Clé étrangère	467
C. Opérations sur la base de données	468
1. Création d'une table	468
2. CRUD	469
a. Insertion	469
b. Lecture	470
c. Mise à jour	470
d. Suppression	471
D. Tri et groupement	471
E. Agrégation	473
F. Jointures	474
1. Jointure interne	475
2. Jointure externe	476
3. Union	477
G. Transactions	477
H. Requêtes préparées	478
I. PDO	478
1. Configuration	479
2. Connexion	480
3. Requêtes	481
a. Requêtes normales	481
b. Requêtes préparées	482
c. Récupération des données	484
4. Transactions	485
J. Validation des acquis : questions/réponses	487
K. Travaux pratiques	490
1. Récupération des sources et de la base de données	490
2. Configurer PHP	490
a. Activation de PDO_MYSQL	490
3. Import de la base de données	491
4. Identification de l'utilisateur	491
5. Création d'un produit	495
6. Liste des produits	498

Chapitre 10	Sécurité
A. Configuration	502
1. PHP	502
a. Gestion des erreurs	502
b. Restriction	503
2. Serveurs	504
a. PHP en tant que binaire CGI	504
b. PHP en tant que module Apache	505
c. Stockage des données	506
d. Sécurisation des données	506
B. Validation de données	507
1. Principe	507
2. Validation par liste blanche	507
3. Validation par liste noire	508
4. Validation par nettoyage	508
5. Moyens de validation	508
a. Filtres/Validateurs	508
b. Validateurs de chaînes de caractères	509
C. XSS	510
1. Principe	510
2. Contre-mesures	510
D. CSRF	512
1. Principe	512
2. Contre-mesures	513
E. Injection	514
1. SQL	514
a. Principe	514
b. Contre-mesures	515
2. Code	516
a. Principe	516
b. Contre-mesures	517
3. Courriers électroniques	519
a. Principe	519
b. Contre-mesures	520
F. Session	521
1. Principe du vol de session	521
2. Principe de la fixation de session	521
3. Contre-mesures	522

G. Chiffrement et algorithme de hachage	523
1. Principes de chiffrement	523
2. Fonctions de hachage	523
a. Fonction md5	524
b. Fonction sha1	524
c. Fonction crypt	524
3. API de hachage de mot de passe	525
a. Constantes de l'API	525
b. Fonctions de l'API	526
H. Chargement des fichiers	528
1. Principe de chargement	528
2. Validation des fichiers chargés	529
I. Validation des acquis : questions/réponses	529
J. Travaux pratiques	532
1. Configurer PHP	532
2. Sécuriser une application	534
a. Récupération des sources et de la base de données	534
b. Import de la base de données	534
c. Mise en place d'une contre-mesure XSS	534
d. Mise en place d'une contre-mesure CSRF	539
e. Sécurisation d'une requête SQL	542
f. Sécurisation d'inclusion de fichiers	545
g. Sécurisation d'injection de courrier électronique	547
h. Sécurisation des sessions	550
i. Chiffrement du mot de passe utilisateur	556
j. Sécurisation du chargement de fichiers	557

Chapitre 11

Fonctionnalités web

A. Session	562
1. Définition	562
2. Constantes	563
3. Variable superglobale	563
4. Fonctions	563
B. Formulaires	566
1. Définition	566
2. Éléments	566
3. Variables superglobales	567

4. Encodage	568
a. Encodage des données de sortie	568
b. Encodage des données d'une URL	568
C. Envoi de fichiers	569
1. Principe	569
2. Configuration	569
3. Variable superglobale	570
4. Manipulations	571
D. Cookies	572
E. En-têtes et code HTTP	576
1. En-têtes HTTP	576
2. Code HTTP	578
a. Nomenclature	578
b. Fonction	578
F. Authentification HTTP	579
G. Validation des acquis : questions/réponses	581
H. Travaux pratiques	584
1. Récupération des sources et de la base de données	584
2. Configurer PHP	584
a. Limitation du chargement de fichiers	584
3. Import de la base de données	585
4. Administration	586
5. Identification de l'utilisateur	588
6. Chargement du CV	591
 Tableau des objectifs	 595
Index	599

Chapitre 5

A. Types de tableaux	202
B. Informations	207
C. Remplissage	210
D. Gestion de la pile	214
E. Itérations	219
F. Recherche	224
G. Extraction	227
H. Tri.	229
I. Comparaison	233
J. Fusion et fission	236
K. SPL, objets comme tableaux.	242
L. Validation des acquis : questions/réponses	244
M. Travaux pratiques	249

Prérequis

Pour bien comprendre ce chapitre, il est nécessaire d'avoir assimilé les bases (cf. chapitre Bases PHP) et les fonctions anonymes (cf. chapitre Fonctions).

Objectifs

À la fin de ce chapitre, vous serez en mesure de :

- ☒ Manipuler des tableaux.
- ☒ Comparer des tableaux.
- ☒ Appliquer un traitement sur tous les éléments d'un tableau.
- ☒ Trier et filtrer les éléments d'un tableau.
- ☒ Réaliser des fusions et des divisions de tableaux.
- ☒ Extraire et remplacer des portions de tableaux.
- ☒ Comprendre le fonctionnement interne des fonctions de tableaux.
- ☒ Résoudre des problèmes complexes à l'aide des tableaux.

A. Types de tableaux

Un tableau en PHP est un type de données qui associe une clé à une valeur.

Concrètement, une clé est définie par une valeur numérique (clé à indexation numérique) ou une chaîne de caractères (clé associative). La valeur quant à elle peut être de n'importe quel type.

☞ En interne, ces deux types de clés n'étant pas différenciés par PHP, les tableaux peuvent être constitués à la fois des unes et des autres.

Un tableau se déclare soit avec la structure du langage `array` soit, depuis PHP 5.4, à l'aide de la notation courte : crochet ouvrant, crochet fermant (`[]`). Chaque paire clé/valeur y figurant doit être séparée des autres par une virgule.

Syntaxes

```
array array([ int|string $key => mixed $args [, $... ]])  
array [ [ int|string $key => mixed $args [, $... ] ]];
```

Exemple d'initialisation de tableaux

```
<?php  
$tableau = array(  
    0 => 1,  
    1 => 'PHP5',  
    'est' => 'formidable',  
);  
  
// Depuis PHP 5.4
```

```
$tableau2 = [
    0 => 1,
    1 => 'PHP5.5',
    'est' => 'fantastique',
];
```

☞ La virgule après le dernier élément n'est pas obligatoire, mais il est communément admis de la faire figurer afin de rendre plus facile l'ajout futur d'éléments lors de la déclaration.

1. Tableaux à indexation numérique

Les tableaux à indexation numérique sont des tableaux dont chaque clé est un nombre entier positif ou négatif.

Lors de la déclaration d'un tableau, la clé étant optionnelle, si celle-ci n'est pas fournie, le moteur PHP utilisera une clé équivalente à l'incrément de l'entier le plus grand déjà utilisé en tant que clé.

Si plusieurs clés fournies lors de la déclaration sont identiques, alors PHP résout le conflit en ne conservant que la dernière valeur fournie pour cette clé.

Exemple de clé numérique

```
<?php
$tableau = array(
    0 => 'Je',
    1 => 'suis',
    2 => 'un tableau',
    3 => 'à',
    4 => 'indexation',
    5 => 'numérique',
);
```

Exemple de tableau contenant des clés non définies

```
<?php
$tableau = array(
    5 => 'Les',
    4 => 'clés',
    'sont',
    20 => 'optionnelles',
    '!',
);

var_dump($tableau);

/* Affiche :
array(5) {
    [5]=> string(3) "Les"
    [4]=> string(5) "clés"
    [6]=> string(4) "sont"
    [20]=> string(12) "optionnelles"
```

```

    [21]=> string(1) "!"
}
*/

```

PHP affecte automatiquement les index s'ils ne sont pas fournis. La valeur *'sonf'* n'a pas d'index défini à ce moment-là. L'index le plus grand est 5 (5 => *'Les'*), la valeur est donc incrémentée puis associée à l'élément (6 => *'sonf'*) ; même principe pour la valeur *'!'* qui se voit associer l'index 21.

Exemple de tableau contenant des clés en conflit

```

<?php
$stableau = array(
    'F',
    'A',
    2 => 'C',
    1 => 'R',
    2 => 'E',
    'B',
    3 => 'D',
);

var_dump($stableau);

/* Affiche :
array(4) {
    [0]=> string(1) "F"
    [1]=> string(1) "R"
    [2]=> string(1) "E"
    [3]=> string(1) "D"
}
*/

```

Cet exemple montre comment PHP résout les conflits de clé. En traitant les éléments dans l'ordre, la valeur *'F'* se voit affecter l'index 0, *'A'* la valeur 1 et enfin *'B'* la valeur 3. Pour chaque clé en conflit, PHP ne conserve que la dernière valeur associée donc la valeur *'R'* écrase la valeur *'A'* à l'index 1, *'E'* écrase *'C'* pour l'index 2 et enfin *'D'* écrase *'E'* pour l'index 3.

2. Tableaux associatifs (à clés littérales)

Un tableau associatif est un tableau dont les clés sont représentées par une chaîne de caractères (même vide). Chaque clé est optionnelle et unique.

Exemple de clé littérale

```

<?php
$stableau = array(
    'PHP' => 'PHP Hypertext Processor',
    'HTML' => 'Hypertext Markup Language',
    'XML' => 'eXtensible Markup Language',
    'CSS' => 'Cascading Style Sheets',

```

```
'SOAP' => 'Simple Object Access Protocol',
'REST' => 'REpresentational State Transfer',
);
```

Exemple avec conflit et clé non définie

```
<?php
$tableau = array(
    'A' => 1,
    'B' => 2,
    3,
    '' => 4,
    'A' => 5,
    'C' => 6,
);

var_dump($tableau);

/* Affiche :
array(5) {
    ["A"]=> int(5)
    ["B"]=> int(2)
    [0]=> int(3)
    [""]=> int(4)
    ["C"]=> int(6)
}
*/
```

La clé littérale 'A' est fournie deux fois lors de la déclaration du tableau, la seconde valeur écrase donc la première et le tableau contient au final cinq éléments.

3. Transtypage de clé

Lors de la déclaration ou de l'affectation dynamique d'une entrée d'un tableau, il est possible que la clé spécifiée ne soit pas un entier. PHP ne l'empêche pas, il le tolère mais convertit la clé soit en entier décimal soit en chaîne de caractères selon des règles bien précises.

Liste des transtypes possibles

Type de clé fournie	Type de clé utilisée
Nombre à virgule flottante, par exemple 4,5	Nombre entier représentant la partie entière, ici 4
Nombre entier non décimal, par exemple 0x66	Équivalent entier décimal, ici 102
Chaîne de caractères représentant un nombre entier valide, par exemple '7'	Nombre entier correspondant, ici 7
chaîne de caractères ne représentant pas un entier valide, par exemple '4,5' ou '08'	Chaîne fournie, ici '4,5' ou '08'
La valeur NULL	Chaîne de caractères vide

Type de clé fournie	Type de clé utilisée
Un booléen, par exemple <code>FALSE</code>	Nombre entier résultant d'un transtypage classique, ici <code>0</code>

☞ Si un tableau ou un objet est fourni en tant que clé, PHP lèvera une erreur de niveau `E_WARNING`,
Warning: Illegal offset.

Exemple de transtypage de clé

```
<?php
$tableau = array(
    2.2 => 'A',
    3,3 => 'B',
    4. => 'C',
    05 => 'D',
    0x6 => 'E',
    '7' => 'F',
    '08' => 'G',
    '9,9' => 'H',
    true => 'I',
    false => 'J',
);

var_dump($tableau);
/* Affiche :
array(10) {
    [2]=> string(1) "A"
    [3]=> string(1) "B"
    [4]=> string(1) "C"
    [5]=> string(1) "D"
    [6]=> string(1) "E"
    [7]=> string(1) "F"
    ["08"]=> string(1) "G"
    ["9,9"]=> string(1) "H"
    [1]=> string(1) "I"
    [0]=> string(1) "J"
}
*/
```

Cet exemple illustre les règles de transtypage de clé listées dans le tableau ci-dessus.

4. Tableaux multidimensionnels

Un tableau multidimensionnel n'est ni plus ni moins qu'un tableau contenant d'autres tableaux, eux-mêmes contenant ou pas d'autres tableaux et ainsi de suite sur une infinité de niveaux.

Il ne possède pas de propriétés particulières mais peut s'avérer particulièrement pratique pour représenter une matrice par exemple.