

Développement serveur en PHP

IO2

Internet et outils

Cristina Sirangelo

IRIF, Université Paris Diderot

cristina@irif.fr

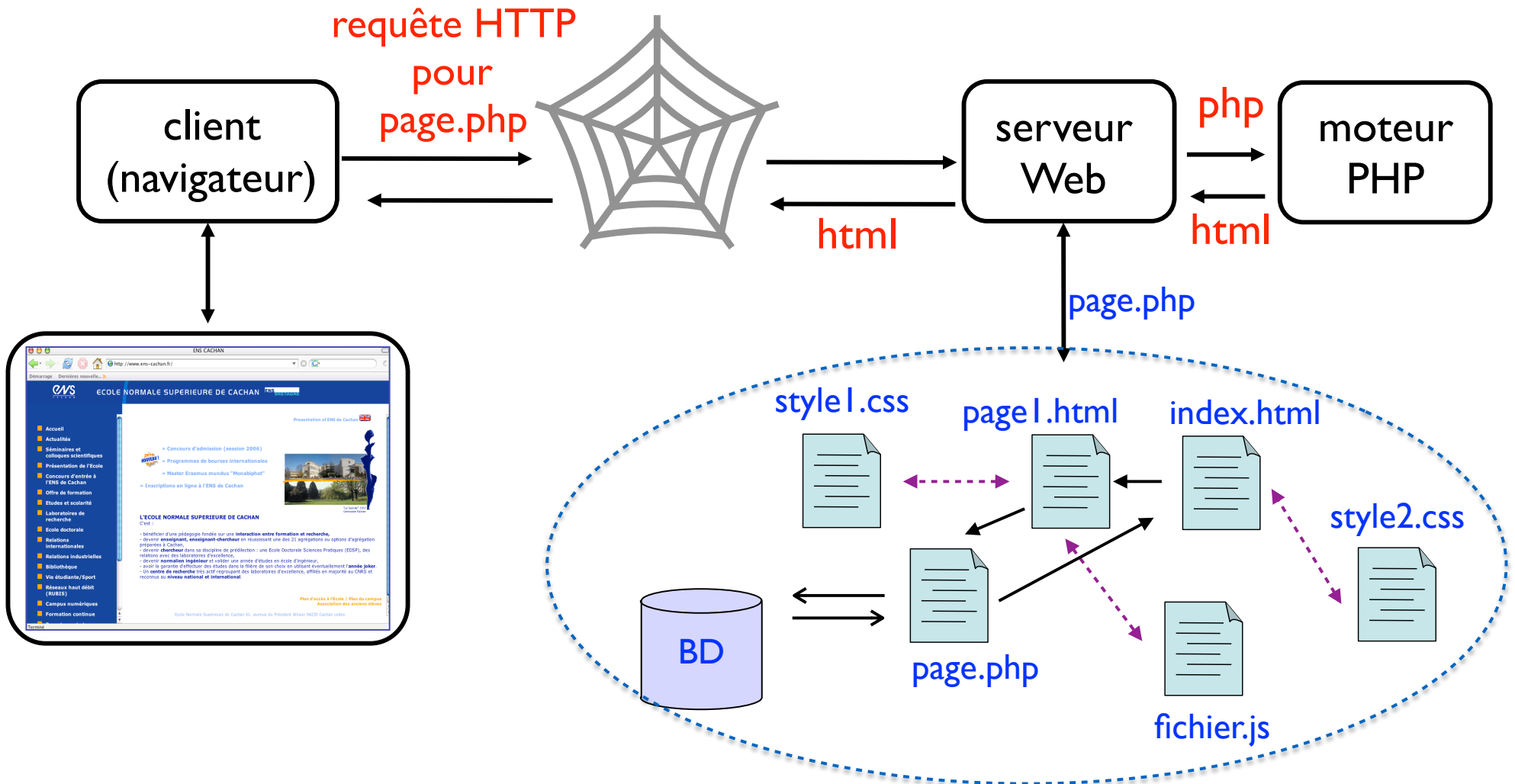
Rappel : structure de base d'une application Web

- Un ensemble de programmes écrits dans un langage de script (e.g. **PHP**) exécutables par le serveur Web, qui implémentent la **logique de l'application** et génèrent des documents HTML dynamiques (**HTML dynamique coté serveur**)
- Eventuellement un ensemble de **documents HTML** statiques
- HTML statique et dynamique, potentiellement enrichi par
 - des **feuilles de style (CSS)** pour définir le style des pages
 - des programmes écrits dans un langage (typiquement **Javascript**) exécutables par le navigateur, qui s'occupent de dynamiser le contenu de la page Web une fois chargée par le navigateur (**HTML dynamique coté client**)
- Possiblement une **base de données** pour le stockage des données d'intérêt de l'application
- Tous ces documents et données : accessibles par un **serveur Web**

PHP est une des langages les plus utilisés pour le développement d'applications Web
(du **PHP pur** au **frameworks**)

Exécution du code coté serveur

- Pour réaliser la logique de l'application, du code est exécuté coté serveur afin de générer les ressources demandées
- Le serveur Web s'appuie sur un moteur de script pour l'interprétation de ce code



PHP Hypertext Preprocessor

- Langage de programmation à la syntaxe proche du Java
- Langage de script
 - ▶ S'utilise et s'exécute côté serveur pour produire un document HTML à renvoyer au client
 - ▶ Imbriqué avec le document HTML
 - ▶ de nombreux modules d'interface avec d'autres outils dont notamment les serveurs de gestion de bases de données (e.g. MySQL)
- Extension des fichiers : `.php`

Script PHP

- Un script PHP est un document incluant des fragments de HTML et des blocs d'instructions PHP

Blocs PHP
délimités par les
pseudo-balises
<?php et ?>

```
<!DOCTYPE html>
<html>
<head> <title> Un script PHP </title></
head>
<body>
```

```
<?php ... ..
```

```
... ..
```

```
?>
```

```
<p> un paragraphe </p>
```

```
<?php... ..
```

```
?>
```

```
</body>
```

```
</html>
```

Script PHP

- Seulement les blocs PHP sont exécutés et ils renvoient du HTML

Page HTML
générée par
le script et
renvoyée au
client



```
<!DOCTYPE html>
```

```
<html>
```

```
<head> <title> Un script PHP </title></head>
```

```
<body>
```

HTML

```
<p> un paragraphe </p>
```

HTML

```
</body>
```

```
</html>
```

Output PHP

- Instruction PHP de base pour produire une chaîne de caractères dans de document :

```
echo ("...") ;
```

- ou avec plusieurs arguments :

```
echo ("...", "...", "...") ;
```

- parenthèses optionnelles
- La chaîne de caractères peut comporter :
 - ▶ des balises HTML
 - ▶ des variables PHP qui seront remplacées par leur valeur

Exemple : fichier côté serveur

```
<html>
<head><title>Essai de PHP</title></head>
<body>
<hr>
Nous sommes le <?php echo date ("j / m / Y"); ?>
<hr>
</body>
</html>
```

`date()` est une fonction PHP (prédéfinie)

Exemple : fichier reçu côté client

```
<html>  
<head><title>Essai de PHP</title></head>  
<body>  
<hr>  
Nous sommes le 12 / 02 / 2021  
  
<hr>  
</body>  
</html>
```

Deuxième exemple

```
<body>
<h1>Un essai de PHP</h1>
<?php
echo "Client : ", $_SERVER["HTTP_USER_AGENT"];
echo "<br><br>Serveur : ";
echo $_SERVER["SERVER_NAME"], "<br>";
?> ...
```

- 3 commandes “echo” séparées par des ;
- Arguments de “echo” séparés par des ,
- `$_SERVER` : une variable PHP prédéfinie (un tableau)
 - contient des information sur le serveur, le client qui demande la page, la requête HTTP, etc.
 - les entrées de ce tableau sont créées par le serveur web et sont disponibles dans tous les scripts (plus de détails plus loin...)

Deuxième exemple (suite)

```
<body>
<h1>Un essai de PHP</h1>
<?php
echo "Client : ", $_SERVER["HTTP_USER_AGENT"];
echo "<br><br>Serveur : ";
echo $_SERVER["SERVER_NAME"], "<br>";
?>
<h3>J'ai réussi mon premier programme PHP
<?php echo "en date du : ", date("d / m / y"), " !" ?>
</h3>
</body>
```

On peut insérer plusieurs séquences PHP dans un fichier

Chaînes entre guillemets : "..."

Code reçu par le navigateur

```
<html>
<head><title>Deuxième exemple de PHP</title></head>
<body>
<h1>Un essai de PHP</h1>
Client : Mozilla/5.0 (Macintosh; Intel Mac OS X 10_13_4) AppleWebKit/
537.36 (KHTML, like Gecko) Chrome/88.0.4324.150 Safari/537.36
<br /><br />Serveur : localhost<br />
<h3>J'ai réussi mon premier programme PHP
  en date du : 12 / 02 / 19 !</h3>
</body></html>
```

En rouge : les parties générées par le code PHP

localhost/io2/04-client-server.php

Rôle du serveur

- Lorsqu'un document HTML est demandé par le client, le serveur se “contente” de trouver le fichier contenant le document correspondant à l'URL et de le renvoyer
- Lorsqu'un script PHP est demandé, le serveur doit d'abord analyser les documents pour :
 - ▶ trouver toutes les occurrences de `<?php... ?>`
 - ▶ faire interpréter les commandes PHP par un interpréteur PHP
 - ▶ substituer les chaînes `<?php... ?>` par le résultat de l'interprétation

PHP

Éléments du langage

Aide, manuel PHP

- Le monde PHP est (très) vaste et (très) riche
- Ce cours n'a pour but que de vous donner quelques bases,
- Si vous cherchez une réponse à une question,
- Si vous cherchez une fonction, une syntaxe,
- Consultez :
 - ▶ Site officiel : <http://www.php.net/>
 - ▶ Pages manuels en français : <https://www.php.net/manual/fr/>
 - ▶ <http://fr.wikipedia.org/wiki/PHP>

Variables

- Noms de variables :
 - ▶ Commencent par \$
 - ▶ Longueur quelconque
 - ▶ Composés de lettres, de chiffres, et _
 - ▶ Lettre ou _ après le \$
 - ▶ Sensibles à la casse (majuscules et minuscules ne sont pas identiques)
- Exemples
 - ▶ \$ma_var, \$Ma_Var, \$mon_1er_nom, \$_une_autre

Variables

- Créer des variables (pas de déclaration de type!) :

```
$note = 19;  
$coeff = 1.53;  
$nom = "Roger Rabbit";  
echo($note);  
echo " Note : ", $note, " Nom : ", $nom,  
      " c : ", $coeff;
```

- Une variable pas définie (`$coeff`) n'a pas de valeur (pas affichée par `echo`)
- Selon les réglages de l'interpréteur PHP, l'utilisation d'une variable non-définie peut générer un message d'alerte :

Notice: Undefined variable: coeff in xxx.php on line 12

Variables

- Supprimer des variables:

- `unset($note);`

- `unset($note, $coeff, $nom);`

- En PHP une variable peut être :

- définie avec une valeur : `$var = "toto";`

- définie et vide : `$var1 = "";` `$var2 = 0;` `$var3 = false;` ...

- non définie : `$var;` *variable sans valeur*

- Vérifier le contenu d'une variable :

- `isset($var)` // Vrai si définie, même vide

- `empty($var)` // Vrai si vide, ou pas définie

Variables : Exemple

```
$note = 19;
```

```
$coeff;
```

```
$nom = "";
```

```
echo
```

```
isset($note), // vrai
```

```
isset($nom), //vrai
```

```
isset($newvar), //faux
```

```
isset($coeff), //faux
```

```
empty($coeff), // vrai
```

```
empty($nom), //vrai
```

```
empty($newvar); //vrai
```

Types de données

- Il existe plusieurs types de base:
 - ▶ Entier (integer) : `$note = 1728;`
 - ▶ Nombre à virgule flottante (float ou double) : `$temp = 37.6;`
 - ▶ Chaîne de caractères (string) : `$nom = "personne";`
 - ▶ Booléen (boolean) : `$condition = TRUE;` `$condition = FALSE;`
- ainsi que des types composés : Tableaux (`array`), Objet (`object`), ...
- Pas de déclaration de type : le type d'une variable est déterminé par son "contenu"
- La même variable peut prendre des types différents au cours de son cycle de vie :

```
$note = 17;    //$note est de type entier
```

```
$note = "très bien"; //$note est maintenant de type string
```

Vérifier le type d'une variable

`is_int($var)` // Vrai si type entier

`is_float($var)`

`is_string($str)`

...

`gettype($var)` renvoie le type de la variable `$var`

Conversion de type

```
$entier = (integer)$var2;
```

```
$chaine = (string)$var3;
```

```
$flottant = (float)$var4;
```

Attention, on ne peut pas convertir n'importe quelle chaîne en entier (0 si impossible)

```
$entier = (int)"1597"; // OK, $entier = 1597
```

```
$entier = (int)"mille deux cents"; // non! $entier = 0
```

```
$entier = (int)"1597 mille deux cents"; // OK!
```

Exemple

<?php

```
$entier1 = 4; $entier2 = 44;  
$chaine = "2021 une bonne année!";  
$string = "une bonne année 2021";  
  
$entier1 = $chaine; $entier2 = $string;  
  
echo "entier1 : ", $entier1, "<br />",  
     "entier2 : ", $entier2, "<br />";  
  
$entier1 = (int)$chaine; $entier2 = (int)$string;  
  
echo "entier1 : ", $entier1, "<br />",  
     "entier2 : ", $entier2, "<br />";
```

?>

Constantes

```
define("nom_cste", valeur_cste);
```

Ne doivent pas correspondre aux mots-clés de PHP

Exemple :

```
define("USD_TO_EUR", 0.929);  
echo "5 dollars valent : ", 5 * USD_TO_EUR,  
    "euros";
```

⇒ 5 dollars valent : 4.64 euros

Commentaires

- Insérer des commentaires en PHP:

```
/* ceci est un  
commentaire */  
  
// commentaire jusqu'à la fin de la ligne  
  
# Ceci également
```

Calculs arithmétiques

Opérateurs :

+ - * / %

`$cpt++;` // équivalent à `$cpt = $cpt + 1;`

`$cpt--;` // équivalent à `$cpt = $cpt - 1;`

`$cpt /= 2;` `$cpt += 77;` `$cpt -= 2;` `$cpt *= 5;`

Quelques fonctions :

`sqrt(64)` // => 8 racine carrée

`ceil(27.68)` // => 28 entier immédiatement supérieur

`floor(27.9)` // => 27 entier immédiatement inférieur

`round(24.62)` // => 25

Opérateurs et types

- Attention : certains opérateurs peuvent changer automatiquement le type d'une variable :

```
$cpt = 7; echo gettype($cpt), "<br>";
```

```
$cpt /= 2; echo gettype ($cpt), "<br>";
```

```
$cpt = 8; echo gettype($cpt), "<br>";
```

```
$cpt /= 2; echo gettype ($cpt), "<br>";
```

```
$str = "10";echo gettype ($str), "<br>";
```

```
$str += 5; echo gettype ($str), "<br>";
```

- Remarque : / sur un entier n'est pas la division entière !

Chaînes de caractères

- Deux formes : `'...'` et `"..."`
- Différence : `"..."` admet interpolation des variables

Chaînes de caractères

- Deux formes : `'...'` et `"..."`
- Différence : `"..."` admet interpolation des variables
 - ▶ une variable présente dans `"..."` sera remplacée par sa valeur

Chaînes de caractères

- Deux formes : '...' et "..."
- Différence : "... " admet interpolation des variables
 - ▶ une variable présente dans "... " sera remplacée par sa valeur
`$var = "jean"`

Chaînes de caractères

- Deux formes : '...' et "..."
- Différence : "... " admet interpolation des variables
 - ▶ une variable présente dans "... " sera remplacée par sa valeur

```
$var = "jean"
```

```
echo "Mon nom est $var <br>"
```

Chaînes de caractères

- Deux formes : '...' et "..."
- Différence : "... " admet interpolation des variables
 - ▶ une variable présente dans "... " sera remplacée par sa valeur

```
$var = "jean"
```

```
echo "Mon nom est $var <br>"
```

⇒ Mon nom est jean

Chaînes de caractères

- Deux formes : `'...'` et `"..."`
- Différence : `"..."` admet interpolation des variables
 - ▶ une variable présente dans `"..."` sera remplacée par sa valeur

```
$var = "jean"
```

```
echo "Mon nom est $var <br>"
```

⇒ Mon nom est jean

```
echo 'Mon nom est $var <br>'
```

Chaînes de caractères

- Deux formes : `'...'` et `"..."`
- Différence : `"..."` admet interpolation des variables
 - ▶ une variable présente dans `"..."` sera remplacée par sa valeur

```
$var = "jean"
```

```
echo "Mon nom est $var <br>"
```

⇒ Mon nom est jean

```
echo 'Mon nom est $var <br>'
```

⇒ Mon nom est \$var

Chaînes de caractères

- Deux formes : `'...'` et `"..."`
- Différence : `"..."` admet interpolation des variables
 - ▶ une variable présente dans `"..."` sera remplacée par sa valeur

```
$var = "jean"
```

```
echo "Mon nom est $var <br>"
```

⇒ Mon nom est jean

```
echo 'Mon nom est $var <br>'
```

⇒ Mon nom est \$var

- Les guillemets doubles doivent être échappés dans `"..."` mais pas dans `'...'`
- Les guillemets simples doivent être échappés dans `'...'` mais pas dans `"..."`

```
$str = "Cette chaîne n'a qu'un \" guillemet double";
```

```
$str = 'Cette chaîne "s\'écrit" avec une apostrophe';
```

Chaînes de caractères

- Concaténer des chaînes :

```
$str1 = "Hello";  
$str2 = "World!";  
$salut = $str1 . " " . $str2;  
echo $salut;  
⇒ Hello World!
```

- Taille d'une chaîne de caractères (nombre de caractères): `strlen()`

```
echo strlen("Hello world!");  
⇒ 12
```

- php offre plusieurs fonctions natives pour travailler avec les chaînes de caractères (cf. plus loin)

Tableaux (array)

- Stockent plusieurs valeurs dans une seule variable
- Les tableaux php peuvent stocker des valeurs de différents types
- Deux formes :
 - ▶ indicés : valeurs associés à des indices 0, 1, 2...
 - ▶ associatifs : valeurs associées à des “clefs” (chaînes de caractères), comme une table de hachage
- Si `$t` est un tableau, `count($t)` renvoie le nombre d'éléments de `$t`
- Comme toute autre variable en PHP, un tableau est créé par affectation de son contenu

Tableaux : création

Créer un tableau indicé :

```
$films[0] = "Casablanca";  
$films[1] = "To Have and Have Not";
```

Pas d'obligation de commencer à 0

Créer un tableau associatif :

```
$capitale["FR"] = "Paris";  
$capitale["UK"] = "Londres";  
$capitale["IT"] = "Rome";
```

Indices et clefs peuvent coexister dans le même tableau :

```
$rues[4] = "Rue Lecourbe"  
$rues["4bis"] = "Rue de la Paix"
```

Tableaux : création

Créer sans indice explicite :

```
$livres[] = "Les Misérables";
```

```
$livres[] = "Notre Dame de Paris";
```

```
$livres[] = "Quatre-vingt treize";
```

```
echo $livres[0] ⇒ Les Misérables
```

```
echo $livres[1] ⇒ Notre Dame de Paris
```

```
echo $livres[2] ⇒ Quatre-vingt treize
```

Tableaux : création

Avec le constructeur `array` (en une seule fois) :

- ▶ Tableau indicé :

```
$livres = array("Les Misérables",  
               "Notre Dame de Paris",  
               "Quatre-vingt treize");
```

```
echo $livres[0] ⇒ Les Misérables
```

- ▶ Tableau associatif :

```
$capitales = array("FR"=> "Paris", "UK"=> "Londre", "IT"=> "Rome");
```

- ▶ Tableau mixte :

```
$rues = array(4 => "Rue Lecourbe", "4bis" => "Rue de la Paix");
```

- ▶ Indices implicites :

```
$rues = array(4 => "Rue Lecourbe", "Rue de la Paix");
```

```
echo $rues[5] ⇒ Rue de la Paix
```

```
$rues = array("4bis" => "Rue de la Paix", "Rue Lecourbe");
```

```
echo $rues[0] ⇒ Rue de la Paix
```


Tableaux : création

Intervalles :

```
$annees = range(1970, 2038);
```

On peut aussi créer des tableaux multi-dimensionnels (tableaux de tableaux)

```
$personnes = array  
(  
    array("nom"=>"Dupont", "prenom" => "Jean"),  
    array("nom"=>"Portier", "prenom" => "Louis", "age" => 39)  
);
```

```
echo $personnes[1]["nom"]
```

⇒ Portier

Tableaux : création

Intervalles :

```
$annees = range(1970, 2038);
```

On peut aussi créer des tableaux multi-dimensionnels (tableaux de tableaux)

```
$notes = array  
(  
    "Dupont" => array("Exo1" => 12, "Exo2" => 14),  
    "Portier" => array("Exo1"=>16, "Exo2" => 18, "Exo3" => 15)  
);
```

```
echo $notes["Portier"]["Exo3"];
```

⇒ 15

Tableaux : création

Création de tableaux multi-dimensionnels par affectation (même effet):

```
$personne[0]["nom"] = "Jean";  
$personne[0]["prenom"] = "Dupont";  
$personne[1]["nom"] = "Louis";  
$personne[1]["prenom"] = "Portier";  
$personne[1]["age"] = 39;
```

```
$notes["Dupont"]["Exo1"] = 12; $notes["Dupont"]["Exo2"] = 14;  
$notes["Portier"]["Exo1"] = 16; $notes["Portier"]["Exo2"] = 18;  
$notes["Portier"]["Exo3"] = 15;
```

Tableaux : impression

```
print_r($livres)
```

```
⇒ Array ( [0] => Les Misérables  
          [1] => Notre Dame de Paris  
          [2] => Quatre-vingt treize )
```

```
var_dump($livres)
```

```
⇒ array(3) { [0]=> string(14) "Les Misérables"  
             [1]=> string(19) "Notre Dame de Paris"  
             [2]=> string(19) "Quatre-vingt treize" }
```

Tableaux : modification

- Modification d'une valeur :

```
$livres[1] = "";
```

```
print_r($livres);
```

```
⇒ Array ( [0] => Les Misérables  
          [1] =>  
          [2] => Quatre-vingt treize )
```

- Suppression d'une case :

```
unset($livres[1]);
```

```
print_r($livres);
```

```
⇒ Array ( [0] => Les Misérables  
          [2] => Quatre-vingt treize )
```

Tableaux : parcours

Une variante de la boucle `for` permet de parcourir aisément les tableaux :

```
foreach($array as $var) {  
    instructions  
}
```

- une itération pour chaque case du tableau
- à l'itération `i`, `$var` prend la valeur de la `i`-ème case du tableau `$array`
- s'applique aussi bien aux tableaux indicés qu'aux tableaux associatifs

Tableaux : parcours

- Exemple avec tableau indicé :

```
$livres = array("Les Misérables",  
               "Notre Dame de Paris",  
               "Quatre-vingt treize");  
  
foreach($livres as $titre) {  
    echo $titre." ";  
};
```

⇒ Les Misérables Notre Dame de Paris Quatre-vingt treize

- Exemple avec tableau associatif :

```
$capitales = array('FR'=> "Paris", 'UK'=> "Londre", 'IT'=> "Rome");  
  
foreach($capitales as $ville) {  
    echo $ville." ";  
}
```

⇒ Paris Londres Rome

Tableaux : parcours

Pour récupérer à chaque itération à la fois la valeur et l'indice/clef de la case :

```
foreach($array as $index => $var) {  
    instructions  
}
```

- Exemple avec tableau indicé :

```
$livres = array("Les Misérables",  
               "Notre Dame de Paris",  
               "Quatre-vingt treize");  
  
foreach($livre as $index => $titre) {  
    echo "Livre #", $index, " : $titre<br/>";  
}
```

```
⇒ Livre #0 : Les Misérables  
   Livre #1 : Notre Dame de Paris  
   Livre #2 : Quatre-vingt treize
```


Tableaux : parcours

Pour récupérer à chaque itération à la fois la valeur et l'indice/clef de la case :

```
foreach($array as $index => $var) {  
    instructions  
}
```

- Exemple avec tableau associatif :

```
$capitales = array('FR'=> "Paris", 'UK'=> "Londre", 'IT'=> "Rome");  
foreach($capitales as $pays => $ville) {  
    echo "capitale de ", $pays, " : $ville<br/>";  
}
```

```
⇒ capitale de FR : Paris  
   capitale de UK: Londres  
   capitale de IT : Rome
```

Tableaux : parcours

Exercice.

Donné un tableau contenant un nombre arbitraire de lignes de la forme :

```
$notes = array  
(  
    "Dupont" => array("Exo1" => 12, "Exo2" => 14),  
    "Portier" => array("Exo1"=>16, "Exo2" => 18, "Exo3" => 15)  
);
```

Calculer un tableau associatif contenant pour chaque étudiant, la moyenne de ses notes ;

l'afficher d'abord avec `print_r()`; ensuite comme un tableau HTML

Tableaux : parcours

Par manipulation d'un pointeur interne au tableau :

```
current($array); // renvoie la valeur sous le pointeur
```

```
next($array); previous($array);
```

```
//avance/recule le pointeur et renvoie la valeur sous le pointeur  
après le déplacement. Renvoie false s'il n'est pas possible  
d'avancer/reculer
```

```
reset($array); end($array);
```

```
//positionne le pointeur sur le premier/dernier élément du tableau
```

Tableaux : parcours

Par manipulation d'un pointeur interne au tableau :

```
current($array); // renvoie la valeur sous le pointeur
```

```
next($array); previous($array);
```

```
//avance/recule le pointeur et renvoie la valeur sous le pointeur  
après le déplacement. Renvoie false s'il n'est pas possible  
d'avancer/reculer
```

```
reset($array); end($array);
```

```
//positionne le pointeur sur le premier/dernier élément du tableau
```

Exemple :

```
$livres = array("Les Misérables", "Notre Dame de Paris",  
               "Quatre-vingt treize");
```

```
reset($livres);
```

```
echo current($livres); // ⇒ Les Misérables
```

```
echo next($livres); // ⇒ Notre Dame de Paris
```

```
echo next($livres); // ⇒ Quatre-vingt treize
```