

Programmation Web Serveur

Chaînes, tableaux, paramètres

D'après les cours de Philippe Renevier



Fabien Givors

Université de Nice Sophia Antipolis

Département Informatique

fabien.givors@unice.fr

CHAÎNES DE CARACTÈRE

CHAÎNES DE CARACTÈRES

- `$chaine = "une chaine de caractère";`
- `strlen($chaine)` pour connaître la taille
- `$chaine{$i}` : le (i+1) ième caractère
- `.` pour concaténer (ajouter deux chaines)
- array **explode** (string \$delimiter , string \$string)
 - Retourne un tableau découpant \$string en fonction des occurrences de \$delimiter
 - Exemple `explode(" / " , "http://deptinfo.unice.fr/~fgivors/index ")`

donnera :

0	http:
1	
2	www.i3s.unice.fr
3	~fgivors
4	index

CHAÎNES DE CARACTÈRES

- Remplacement des variables par leurs valeurs quand elles sont entre " ... "
- Si `$chaine = "une chaîne de caractère"`,
 - `$msg = "<p>la variable \$chaine vaut : $chaine</p>";`
 - `$msg` contient le texte «la variable `$chaine` vaut : une chaîne de caractère»
 - Caractères spéciaux : pour les voir tels quels il faut les précéder d'un \
 - `\$` pour avoir un caractère '\$'
 - `\n` : retour à la ligne ; `\t` : tabulation
- Fonction `print` de php : pour afficher (générer du html) :
 - `print($msg) ; // le contenu de $msg sera visible dans la page web`
 - `echo $msg ; // équivalent`
 - `echo "$msg" ; // équivalent`

LES VARIABLES DANS LES CHAÎNES DE CARACTÈRES

Entourez le nom de la variable avec des accolades pour indiquer explicitement son nom.

```
<?php
```

```
$boisson = 'vin';
```

```
// Correct, car "," n'est pas autorisé dans les noms de variables
```

```
echo "Du $boisson, du pain et du fromage!";
```

```
// Pas correct, car 's' peut faire partie d'un nom de variable, et  
PHP recherchera alors $boissons
```

```
echo "Il a goûté plusieurs $boissons";
```

```
echo "Il a goûté plusieurs {$boisson} s"; // Correct
```

```
?>
```

LES CARACTÈRES D'ÉCHAPPEMENT POUR DES CHAINES ENTRE " "

- `\n` Saut de ligne
- `\r` Retour chariot
- `\t` Tabulation
- `\\` Le signe '\'
- `\$` Le signe '\$'
- `\"` Un guillemet double
- `\0nn` Une chaîne en octal
- `\xnn` Une chaîne en hexadécimal

FONCTIONS COURANTES

- Sous chaîne de caractère
 - substr
 - string **substr** (string \$string , int \$start [, int \$length])
 - Retourne le segment de **\$string** défini par **\$start** et **\$length**.
 - Ne modifie pas **\$string**
- Recherche dans une chaîne
 - strpos
 - int **strpos** (string \$string , mixed \$recherche)
 - Cherche la position numérique de la première occurrence de **\$recherche** dans **\$string**.

CONVERSION

- Vers une chaîne de caractère
 - (string)
 - fonction `strval()`
 - Conversion automatique (cas d'écho)
 - Un tableau sera converti en « Array » : utilisation de la fonction `var_dump()` ou `print_r()`
- Vers un entier
 - fonction `intval`

TABLEAUX

TABLEAUX

- Un tableau est une suite de valeurs référencées par une unique variable.
- PHP gère dynamiquement la taille des tableaux
- Les tableaux en PHP peuvent être soit *indicés* soit *associatifs*.
- Tableaux indicés : quelques exemples.

```
$tab[0] = " élément 1 "; $tab[1] = "élément 2 "; $tab[2] = 120;
```

- PHP affecte automatiquement un indice à un nouvel élément du tableau. Cet indice est le numéro de la première cellule vide. Donc le code ci-dessous est équivalent au précédent.

```
$tab[] = "élément 1 "; // $tab[0] ! ;
```

```
$tab[] = "élément 2 "; // $tab[1] !
```

```
$tab[] = 120;
```

TABLEAUX

- L'instruction array
 - moyen d'initialiser facilement un tableau..
 - `$tab = array ("élément 1 ", "élément 1 ", 120);`
- Tableaux associatifs
 - Indexation par clef
 - Exemple : tableau de films

Définition classique:

```
$mes["Vertigo"] = "Hitchcock";  
$mes["Sacrifice"] = "Tarkovski";  
$mes["Alien"] = "Scott";
```

Utilisation array:

```
$mes = array (  
    "Vertigo" => "Hitchcock",  
    "Sacrifice" => "Tarkovski",  
    "Alien" => "Scott"  
);
```

TABLEAUX MULTI-DIMENSIONNELS

- Les tableaux indicés et associatifs se généralisent aux tableaux multi-dimensionnels, pour lesquels l'indice, ou la clé, est constituée de plusieurs valeurs. Un tableau à deux dimensions peut être vu comme une table avec lignes et colonnes. Exemple :

```
$tab[0][0] = "En haut gauche";  
$tab[0][1] = "En haut droite";  
$tab[1][0] = "En bas gauche";  
$tab[1][1] = "En bas droite";
```

Utilisation array:

```
$mes = array (  
    "Vertigo" => array ("Alfred", "Hitchcock"),  
    "Sacrifice" => array ("Andrei", "Tarkovski"),  
    "Alien" => array ("Ridley", "Scott")  
);
```

Note :

Dans l'exemple avec array les tableaux imbriqués sont indicés et contiennent chacun deux éléments. `$mes["Vertigo"][1]` est donc la chaîne "Hitchcock".

CONNAITRE LE NOMBRE D'ÉLÉMENT D'UN TABLEAU

- Fonction `count`

- Exemple

- `$tab = array ("élément 1 ", "élément 1 ", 120);`
`count($tab)` vaut 3

- `$mes = array (`
 `"Vertigo" => array ("Alfred", "Hitchcock"),`
 `"Sacrifice" => array ("Andrei", "Tarkovski"),`
 `"Alien" => array ("Ridley", "Scott"));`
`count($mes)` vaut 3
`count($mes["Vertigo"])` vaut 2

UNSET

- ▣ **unset()** détruit les variables spécifiées.
(ne retourne pas de valeurs)

```
<?php
$arr = array(5 => 1, 12 => 2);
$arr[] = 56; // Ceci revient à $arr[13] = 56; à ce
moment du script
$arr["x"] = 42; // Ceci ajoute un nouvel élément avec
l'index "x"
unset($arr[5]); // Ceci efface un élément du tableau
unset($arr); // Ceci efface tout le tableau
?>
```

FONCTIONS UTILES POUR LES TABLEAUX

▣ Pour ré-indexer : `array_values()`

```
<?php
$a = array(1 => 'un', 2 => 'deux', 3 => 'trois');
unset($a[2]);
/* Cela va produire un tableau qui aurait été $a = array(1
   => 'un', 3 => 'trois')
et non pas $a = array(1 => 'un', 2 =>'trois'); */
$b = array_values($a); //Maintenant b est le tableau
   array(1 => 'un', 2 =>'trois')
?>
```

▣ `sort($tab, flag) : tri`

▣ `SORT_REGULAR ; SORT_NUMERIC ; SORT_STRING`

TESTS SUR LES VARIABLES

- ▣ Pour connaître le type :
 - ▣ `gettype()` (retourne une chaîne)
 - ▣ Fonctions qui retournent un booléen :
`is_array()`, `is_float()`, `is_int()`, `is_object()` et `is_string()`.
- ▣ `isset` : pour savoir si elle(s) existe(nt) :
 - ▣ `isset`
 - ▣ retourne vrai ou faux
 - ▣ permet de savoir si la variable « nom » existe
 - ▣ `isset($variable)`
 - ▣ `isset($v1, $v2, $v3)`

UN ÉLÉMENT D'UN TABLEAU DANS UNE CHAÎNE

```
<?php
// Ces exemples sont spécifiques à l'utilisation de tableaux dans une
chaîne
// Lorsque vous êtes hors d'une chaîne, utilisez toujours des guillemets
autour des index de tableau, et n'utilisez pas d'accolades}.

// le tableau
$fruits = array('fraise' => 'rouge', 'banane' => 'jaune');

// Fonctionne mais notez que cela ne fonctionne pas hors d'une chaîne
echo "Une banane est $fruits[banane].";

// Fonctionne
echo "Une banane est {$fruits['banane']}.";

// Ne fonctionne pas
echo "Une banane est {fruits[banane]}.";

// Fonctionne
echo "Une banane est " . $fruits['banane'] . ".";
?>
```

En attendant les formulaires...

PASSAGE DE PARAMÈTRE À UNE PAGE (1)

D'UNE PAGE À UNE AUTRE

- ▣ Chaque page exécutée (ou obtenue) indépendamment
- ▣ Continuité dans la navigation, dans l'interaction
- ▣ Comment partager des valeurs ?
 - ▣ Persistance à long terme (fichiers, bd)
 - ▣ Persistance à court terme (session)
 - ▣ Formulaire (saisie par l'utilisateur)
 - ▣ Paramètre dans l'url et lien spécifique

DANS L'URL

- Ajout à la fin de l'url
- `http://serveur.domaine/chemin/page?variable=valeur`
- Une variable dans une url se définit par
 - Un nom
 - Une valeur (après égal)
- Plusieurs variables possibles
 - Séparation par des & entre les variables
 - <https://www.google.fr/search?q=Fabien+Givors&aq=f&ie=UTF-8>
- Fonctionne dans des liens
``

RÉCUPÉRER LA VALEUR DANS PHP

- `$_GET`
 - Variable super globale
 - <http://www.php.net/manual/fr/language.variables.superglobals.php>
 - Un tableau associatif des valeurs passées au script courant via les paramètres d'URL.
- Les clefs sont les noms des variables de l'url
- Les valeurs sont définies après le signe = (et avant le &)
- Encodage de l'url
 - Décodage avec `urldecode` ou `rawurldecode`
 - Encodage avec `urlencode` ou `rawurlencode`
 - <http://www.faqs.org/rfcs/rfc3986.html>

TESTER SI UNE VARIABLE PASSÉE PAR URL EXISTE

```
$val = "valeur par default";  
if ( isset($_GET["variable"]) )  
{  
    $val = $_GET["variable"];  
    // déjà décodé (encode url comme %20)  
    // faire des tests sur les valeurs...  
    // conversion  
}
```