

# Les Tableaux en PHP

- Un tableau est une collection de paires (key, value), où la clé key est utilisée pour identifier des valeurs dans la collection
- en PHP, la clé peut être n'importe quelle valeur : entier, réel, chaîne ...
- La valeur associée à une clé peut-être un tableau

Key	value
1	"un"
2	"deux"
3	"trois"
"nom"	"Franck"
"age"	45
"genre"	M
"mobile"	0656903424

## ■ Accès dans le tableau

```
$guy = $tab["nom"];  
$one = $tab[1];
```

## ■ Ajout d'une entrée :

```
$tab[4] = "quatre";  
$tab['adress'] = "London";
```

## ■ Suppression d'une entrée :

```
unset($arr[4]);
```

# Création d'un tableau

- expression littérale : [ ... ]
- Avec l'instruction array()
- Par ajout de valeur

```
<?php
// ceci
$fruit = array( 'color' => 'red', 'taste' => 'sweet',
                'name'  => 'apple', 1  => 'cortland',
                2  => 'macintosh' );

// est équivalent à cela
$fruit['color'] = 'red';
$fruit['taste'] = 'sweet';
$fruit['name']  = 'apple';
$fruit[1] = 'cortland';
$fruit[2] = 'macintosh';

$fruit = [ 'color' => 'red', 'taste' => 'sweet',
           'name'  => 'apple', 1  => 'cortland',
           2  => 'macintosh' ];
```

- Un tableau peut stocker tous types de valeurs, y compris un tableau :

```
<?php
$myAddressBook = [
    'john' => [ 'name'=>"john smith",
                'email'=>"js@hotmail.com" ) ],
    'sarah'=> [ 'name'=>"sarah Lenti",
                'email'=>"sl@free.fr" ],
    'quentin'=> [ 'name'=>"Quentin Tarantino",
                  'email'=>"qt@kill-bill.fi" ]
];

// echoing Quentin Tarantino email !!
echo "quentin mail is: $myAddressBook['quentin']
['email'] <br>";
```

- C'est la façon de stocker des tableaux à dimension multiples (ex. des matrices)

- La clé peut être oubliée lors de l'ajout d'une valeur.
  - Choisie automatiquement : plus grande clé entière + 1

```
<?php
$arr = [ 5 => 1, 12 => 2];

$arr[] = 56;    // pareil que $arr[13] = 56;

// ce tableau est le même que ...
$a = [ 5 => 43, 32, 56, "b" => 12 ];

// ... ce tableau
$a = [ 5 => 43, 6 => 32, 7 => 56, "b" => 12 ];
```

# Itération sur les tableaux

- Très courant :
  - Imprimer le contenu d'un tableau
  - Faire des recherches dans un tableau
  - Appliquer un calcul à chaque élément du tableau

## ■ Utiliser la boucle spécialisée FOREACH

```
foreach ($arr as $value) statement  
foreach ($arr as $key => $value) statement
```

- *\$key* et *\$value* sont des variables qui vont successivement stocker les éléments du tableau à chaque itération

```
<?php  
  
$t = [ 'a', 'b', 'c', 'd', 'e'=>'euuuh' ] ;  
  
foreach ($t as $index => $item) {  
    echo 'entrée : ' . $index . ' : ' . $item ;  
}
```

## ■ Affectation de tableaux

- Affectation directe
- Affectation d'une liste de variables

```
<?php
```

```
$a = [ 'r', 'o', 'j', 'v', 'b', 'i', 'v' ] ;
```

```
$tab_colors = $a;
```

```
list($c1,$c2,$c3,$c4,$c5,$c6,$c7) = $a;
```



# Quelques Fonctions sur les tableaux

**int count** ( mixed *var* [, int *mode*])

returns the number of elements in *var*, which is typically an **array**

**array array\_diff** ( array *array1*, array *array2* [, array ...])

**array\_diff()** returns an array containing all the values of *array1* that are not present in any of the other arguments.

**array array\_merge** ( array *array1*, array *array2* [, array ...])

**array\_merge()** merges the elements of two or more arrays together so that the values of one are appended to the end of the previous one

**array array\_reverse** ( array *array* [, bool *preserve\_keys*])

**array\_reverse()** takes input *array* and returns a new array with the order of the elements reversed, preserving the keys if *preserve\_keys* is **TRUE**.

**mixed array\_search** ( mixed *needle*, array *haystack* [, bool *strict*])

Searches *haystack* for *needle* and returns the key if it is found in the array, **FALSE** otherwise.

# Tableaux Prédéfinis (SuperGlobaux)

- **\$\_SERVER** : infos sur le serveur et la requête HTTP courante

**\$\_SERVER[ 'PHP\_SELF' ]**

The filename of the currently executing script

**\$\_SERVER[ 'SERVER\_NAME' ]**

The name of the server host under which the current script is executing.

**\$\_SERVER[ 'REQUEST\_METHOD' ]**

Which request method was used to access the page; i.e. 'GET', 'HEAD', 'POST', 'PUT'.

**\$\_SERVER[ 'HTTP\_USER\_AGENT' ]**

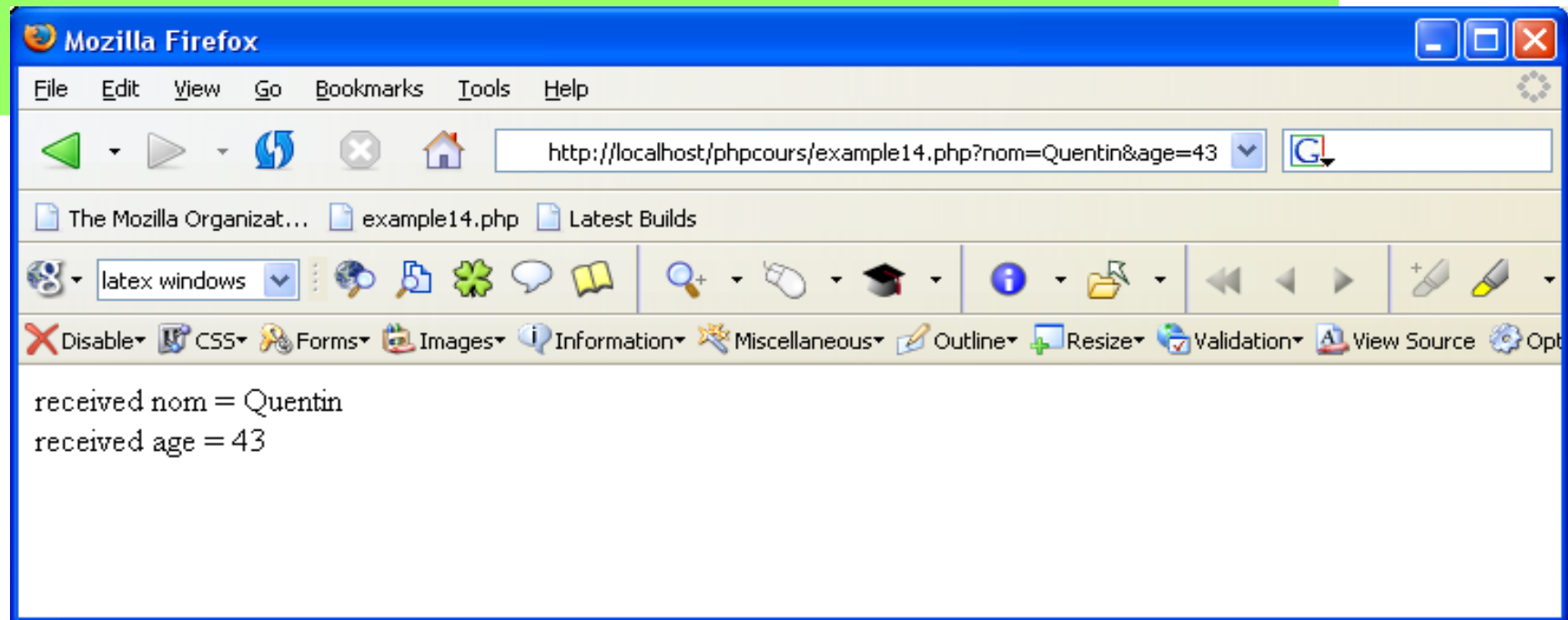
Contents of the User-Agent: header from the current request, if there is one.

**\$\_SERVER[ 'REMOTE\_HOST' ]**

The Host name from which the user is viewing the current page.

- \$\_GET : Valeurs passées avec la méthode GET, dans l'URL

```
<?php
foreach ($_GET as $var=>$val) {
    echo "received $var = $val<br>";
}
```



- Moyen le plus simple de passer des paramètres à un script PHP

# exercices

- TD n°4