

# Développement PHP

## Partie 2 : Bases de PHP

DENIS LOEUILLET – IFA - 2017

# Partie 2 : Bases de PHP

- Variables, types de variables
- Les instructions conditionnelles
- Les boucles
- les types de données composés : tableaux et objets
- Les fonctions
- Les constantes

# Partie 2 : Bases de PHP

## Les tableaux (arrays)

<http://php.net/manual/fr/language.types.array.php>

- Un aspect très important du PHP : les arrays.
- Il s'agit de variables « composées ».
- On peut faire énormément de choses avec les arrays
- Leur utilisation n'est pas toujours très facile.
- Ils sont indispensables à la programmation en PHP
- Un tableau (aussi appelé array) est une variable. Mais une variable un peu spéciale.

# Partie 2 : Bases de PHP

## Les tableaux (arrays)

1. Les deux types de tableaux
2. Parcourir un tableau
3. Rechercher dans un tableau

# Partie 2 : Bases de PHP

## Les tableaux (arrays)

### 1. Les deux types de tableaux

- Jusqu'ici vous avez travaillé avec des variables toutes simples :
  - elles ont un nom et une valeur.
  - Par exemple :

```
1 <?php
2 $prenom = 'Nicole';
3 echo 'Bonjour ' . $prenom; // Cela affichera : Bonjour Nicole
4 ?>
```

- Ce qui peut se matérialiser sous la forme :

Nom	Valeur
\$prenom	Nicole

# Partie 2 : Bases de PHP

## Les tableaux (arrays) : les deux types de tableaux

- il est possible d'enregistrer de nombreuses informations dans une seule variable grâce aux tableaux.
- On en distingue deux types :
  - Les tableaux numérotés
  - Les tableaux associatifs



# Partie 2 : Bases de PHP

## Les tableaux (arrays) : les deux types de tableaux

- Les tableaux numérotés

- Simples à imaginer

- Exemple :

- ✓ \$prenoms est un array : c'est une variable « tableau »
- ✓ Variable de type composée
- ✓ Elle n'a pas qu'une valeur, mais plusieurs (on peut en mettre tant qu'on veut)
- ✓ Les variables sont rangées dans des « cases » différentes
- ✓ Array numéroté
  - ❖ Chaque case est identifiée par un numéro
  - ❖ Ce numéro est appelé « clé »
  - ❖ La numérotation commence toujours à la case n°0

Clé	Valeur
0	François
1	Michel
2	Nicole
3	Véronique
4	Benoît
...	...

# Partie 2 : Bases de PHP

## Les tableaux (arrays) : les deux types de tableaux

- Construire un tableau numéroté

- Pour créer un tableau numéroté en PHP, on utilise généralement la fonction « array »
- Exemple : créer l'array \$prenoms

```
1 <?php
2 // La fonction array permet de créer un array
3 $prenoms = array ('François', 'Michel', 'Nicole', 'Véronique', 'Benoît');
4 ?>
```

- ✓ L'ordre a beaucoup d'importance.
- ✓ Le premier élément (« François ») aura le n°0
- ✓ ensuite Michel le n°1, etc.



# Partie 2 : Bases de PHP

## Les tableaux (arrays) : les deux types de tableaux

- Construire un tableau numéroté

➤ Vous pouvez aussi créer manuellement le tableau case par case :

```
1 <?php
2 $prenoms[0] = 'François';
3 $prenoms[1] = 'Michel';
4 $prenoms[2] = 'Nicole';
5 ?>
```

✓ Si vous ne voulez pas avoir à écrire vous-mêmes le numéro de la case que vous créez, vous pouvez laisser PHP le sélectionner automatiquement en laissant les crochets vides :

```
1 <?php
2 $prenoms[] = 'François'; // Créera $prenoms[0]
3 $prenoms[] = 'Michel'; // Créera $prenoms[1]
4 $prenoms[] = 'Nicole'; // Créera $prenoms[2]
5 ?>
```

## Partie 2 : Bases de PHP

### Les tableaux (arrays) : les deux types de tableaux

- Afficher un tableau numéroté
  - Pour afficher un élément, il faut donner sa position entre crochets après \$prenoms.
  - Cela revient à dire à PHP :
    - « Affiche-moi le contenu de la case n°1 de \$prenoms. »
  - Pour faire cela en PHP, il faut écrire le nom de la variable, suivi du numéro entre crochets. Pour afficher « Michel », on doit donc écrire :

```
1 <?php
2 echo $prenoms[1];
3 ?>
```

- ✓ Michel est en seconde position
- ✓ Il a le numéro 1 car on commence à numéroté à 0

Remarque : Si vous oubliez de mettre les crochets, ça ne marchera pas (ça affichera juste « Array »...). Dès que vous travaillez sur des arrays, vous êtes obligés d'utiliser les crochets pour indiquer dans quelle « case » on doit aller chercher l'information, sinon PHP ne sait pas quoi récupérer.

# Partie 2 : Bases de PHP

## Les tableaux (arrays) : les deux types de tableaux

- Les tableaux associatifs
  - Fonctionnent sur le même principe que les tableaux numérotés
  - Au lieu de numéroté les case, on va les étiqueter en leur donnant à chacune un nom différent.
  - Par exemple :
    - ✓ supposons que je veuille, dans un seul array, enregistrer les coordonnées de quelqu'un :
      - nom
      - Prénom
      - Adresse
      - Ville
      - etc.
    - ✓ Si l'array est numéroté, comment savoir que le n°0 est le nom, le n°1 le prénom, le n°2 l'adresse... ?
    - ✓ C'est là que les tableaux associatifs deviennent utiles.

## Partie 2 : Bases de PHP

### Les tableaux (arrays) : les deux types de tableaux

- Construire un tableau associatif
  - On utilise la fonction array comme pour les tableaux numérotés
  - On va mettre l'étiquette devant chaque information

```
1 <?php
2 // On crée notre array $coordonnees
3 $coordonnees = array (
4     'prenom' => 'François',
5     'nom' => 'Dupont',
6     'adresse' => '3 Rue du Paradis',
7     'ville' => 'Marseille');
8 ?>
```

- ✓ il n'y a qu'une seule instruction (un seul « ; »)
- ✓ On peut tout écrire sur une ligne
- ✓ Plus lisible sur plusieurs lignes
- ✓ On écrit une flèche « => » pour dire « associé à »
  - par exemple : « ville » associé à « Marseille »

Clé	Valeur
prenom	François
nom	Dupont
adresse	3 Rue du Paradis
ville	Marseille

# Partie 2 : Bases de PHP

## Les tableaux (arrays) : les deux types de tableaux

- Construire un tableau associatif

➤ Il est aussi possible de créer un tableau associatif case par case :

```
1 <?php
2 $coordonnees['prenom'] = 'François';
3 $coordonnees['nom'] = 'Dupont';
4 $coordonnees['adresse'] = '3 Rue du Paradis';
5 $coordonnees['ville'] = 'Marseille';
6 ?>
```

# Partie 2 : Bases de PHP

## Les tableaux (arrays) : les deux types de tableaux

- Afficher un tableau associatif

- il suffit d'indiquer le nom de cet élément entre crochets, ainsi qu'entre guillemets ou apostrophes puisque l'étiquette du tableau associatif est un texte.

```
1 <?php
2 echo $coordonnees['ville'];
3 ?>
```

- ✓ ce code affiche « Marseille »
- ✓ Quand utiliser un array numéroté et quand utiliser un array associatif ?
  - les deux types servent à stocker la même chose
  - Les arrays numérotés permettent de stocker une série d'éléments du même type, comme des prénoms. Chaque élément du tableau contiendra alors un prénom.
  - Les arrays associatifs permettent de découper une donnée en plusieurs sous-éléments. Par exemple, une adresse peut être découpée en nom, prénom, nom de rue, ville...



# Partie 2 : Bases de PHP

## Les tableaux (arrays)

### 2. Parcourir un tableau

- On a souvent besoin de savoir ce qu'il contient, pour cela trois moyens sont disponibles :
  - la boucle « for »
  - La boucle « foreach »
  - La fonction « print\_r » (utilisée principalement pour le débogage)

# Partie 2 : Bases de PHP

## Les tableaux (arrays) : parcourir un tableau

- La boucle « for »
  - très simple de parcourir un tableau numéroté avec boucle « for »
    - ✓ tableau numéroté à partir de 0
    - ✓ on peut faire une boucle « for » qui incrémente un compteur à partir de 0

```
1 <?php
2 // On crée notre array $prenoms
3 $prenoms = array ('François', 'Michel', 'Nicole', 'Véronique', 'Benoît');
4
5 // Puis on fait une boucle pour tout afficher :
6 for ($numero = 0; $numero < 5; $numero++)
7 {
8     echo $prenoms[$numero] . '<br />'; // affichera $prenoms[0], $prenoms[1] etc.
9 }
10 ?>
```

- ❖ Quand on écrit `$prenoms[$numero]`, la variable `$numero` est d'abord remplacée par sa valeur. Par exemple, si `$numero` vaut 2, alors cela signifie qu'on cherche à obtenir ce que contient `$prenoms[2]`, c'est-à-dire... Nicole.

# Partie 2 : Bases de PHP

## Les tableaux (arrays) : parcourir un tableau

- La boucle « foreach »

<http://php.net/manual/fr/control-structures.foreach.php>

- Dans la plupart des langages de programmation, le parcours de tableau se réalise à l'aide de boucles.
- PHP dispose quant à lui de sa propre structure de contrôle permettant de parcourir le contenu d'un tableau.
- Il s'agit de la structure `foreach()`.
- C'est une boucle particulière qui avance le pointeur du tableau à chaque itération.
- C'est une boucle « for » spécialisée dans les tableaux

# Partie 2 : Bases de PHP

## Les tableaux (arrays) : parcourir un tableau

- La boucle « foreach »
  - « foreach » va passer en revue chaque ligne du tableau
  - lors de chaque passage, elle va mettre la valeur de cette ligne dans une variable temporaire (par exemple \$element).

```
1 <?php
2 $prenoms = array ('François', 'Michel', 'Nicole', 'Véronique', 'Benoît');
3
4 foreach($prenoms as $element)
5 {
6     echo $element . '<br />'; // affichera $prenoms[0], $prenoms[1] etc.
7 }
8 ?>
```

- ✓ C'est le même code que tout à l'heure mais avec « foreach »
- ✓ À chaque tour de boucle, la valeur de l'élément suivant est mise dans la variable \$element.
- ✓ On peut donc utiliser \$element uniquement à l'intérieur de la boucle afin d'afficher l'élément en cours.
- ✓ L'avantage de « foreach » est qu'il permet aussi de parcourir les tableaux associatifs.

# Partie 2 : Bases de PHP

## Les tableaux (arrays) : parcourir un tableau

- La boucle « foreach »

```
1 <?php
2 $coordonnees = array (
3     'prenom' => 'François',
4     'nom' => 'Dupont',
5     'adresse' => '3 Rue du Paradis',
6     'ville' => 'Marseille');
7
8 foreach($coordonnees as $element)
9 {
10     echo $element . '<br />';
11 }
12 ?>
```

➤ « foreach » va mettre tour à tour dans la variable \$element :

- ✓ le prénom
- ✓ le nom
- ✓ l'adresse
- ✓ la ville

contenus dans l'array \$coordonnees.

# Partie 2 : Bases de PHP

## Les tableaux (arrays) : parcourir un tableau

- La boucle « foreach »
  - On met entre parenthèses :
    - ✓ d'abord le nom de l'array (ici \$coordonnees) ;
    - ✓ ensuite le mot-clé « as » (qui signifie quelque chose comme « en tant que ») ;
    - ✓ enfin, le nom d'une variable que vous choisirez et qui va contenir tour à tour chacun des éléments de l'array (ici \$element).
  - Entre les accolades, on n'utilise donc que la variable \$element.
  - La boucle s'arrête lorsqu'on a parcouru tous les éléments de l'array.
  - Toutefois, avec cet exemple, on ne récupère que la valeur :
    - ✓ Or, on peut aussi récupérer la clé de l'élément.
    - ✓ On doit dans ce cas écrire « foreach » comme ceci :

```
1 <?php foreach($coordonnees as $cle => $element) ?>
```



# Partie 2 : Bases de PHP

## Les tableaux (arrays) : parcourir un tableau

- La boucle « foreach »

```
1 <?php foreach($coordonnees as $cle => $element) ?>
```

- À chaque tour de boucle, on disposera non pas d'une, mais de deux variables :
  - ✓ \$cle, qui contiendra la clé de l'élément en cours d'analyse (« prenom », « nom », etc.) ;
  - ✓ \$element, qui contiendra la valeur de l'élément en cours (« François », « Dupont », etc.).
- Exemple :

```
1 <?php
2 $coordonnees = array (
3     'prenom' => 'François',
4     'nom' => 'Dupont',
5     'adresse' => '3 Rue du Paradis',
6     'ville' => 'Marseille');
7
8 foreach($coordonnees as $cle => $element)
9 {
10     echo '[' . $cle . ']' vaut ' . $element . '<br />';
11 }
12 ?>
```

- Avec cette façon de procéder, vous avez maintenant dans la boucle la clé ET la valeur.
- C'est une boucle très pratique
- On s'en sert régulièrement

# Partie 2 : Bases de PHP

## Les tableaux (arrays) : parcourir un tableau

- Afficher rapidement un « array » avec « print\_r »

<http://php.net/manual/fr/function.print-r.php>

- Parfois, en codant votre site en PHP, vous aurez un array et vous voudrez savoir ce qu'il contient, juste pour votre information.
- Vous pourriez utiliser une boucle « for » ou, mieux, une boucle « foreach ». Mais :
  - ✓ si vous n'avez pas besoin d'une mise en forme spéciale
  - ✓ et que vous voulez juste savoir ce que contient l'array
    - vous pouvez faire appel à la fonction « print\_r ».
    - C'est une sorte de « echo » spécialisé dans les arrays.
    - Cette commande a toutefois un défaut :
      - elle ne renvoie pas de code HTML comme « <br /> » pour les retours à la ligne.
      - Pour bien les voir, il faut donc utiliser la balise HTML « <pre> » ([https://www.w3schools.com/tags/tag\\_pre.asp](https://www.w3schools.com/tags/tag_pre.asp)) qui nous permet d'avoir un affichage plus correct.

# Partie 2 : Bases de PHP

## Les tableaux (arrays) : parcourir un tableau

- Afficher rapidement un « array » avec « print\_r »

```
1 <?php
2 $coordonnees = array (
3     'prenom' => 'François',
4     'nom' => 'Dupont',
5     'adresse' => '3 Rue du Paradis',
6     'ville' => 'Marseille');
7
8 echo '<pre>';
9 print_r($coordonnees);
10 echo '</pre>';
11 ?>
```

- Facile à utiliser
- Ne pas oublier la balise « <pre> » pour un affichage correct
- À n'utiliser que pour le débogage, pendant la création du code afin de visualiser le contenu d'un array

# Partie 2 : Bases de PHP

## Les tableaux (arrays)

### 2. Recherche dans un tableau

- Savoir si un array contient ou non certaines informations
- Trois types de recherche :
  - `array_key_exists` : pour vérifier si une clé existe dans l'array ;
  - `in_array` : pour vérifier si une valeur existe dans l'array ;
  - `array_search` : pour récupérer la clé d'une valeur dans l'array.

# Partie 2 : Bases de PHP

## Les tableaux (arrays) : recherche dans array

- Vérifier si une clé existe dans l'array : « array\_key\_exists »  
<http://php.net/manual/fr/function.array-key-exists.php>
- On a un array, mais on ne sait pas si la clé qu'on cherche s'y trouve
  - on va utiliser la fonction « array\_key\_exists »
  - « array\_key\_exists » va parcourir le tableau pour nous et nous dire s'il contient cette clé
  - On doit d'abord lui donner le nom de la clé à rechercher
  - puis le nom de l'array dans lequel on fait la recherche :

```
1 <?php array_key_exists('cle', $array); ?>
```

## Partie 2 : Bases de PHP

### Les tableaux (arrays) : recherche dans array

- Vérifier si une clé existe dans l'array : « array\_key\_exists »
  - La fonction renvoie un booléen:
    - ✓ true (vrai) si la clé est dans l'array
    - ✓ false (faux) si la clé ne s'y trouve pas
  - Ça nous permet de faire un test facilement avec un « if » :

```
1 <?php
2 $coordonnees = array (
3     'prenom' => 'François',
4     'nom' => 'Dupont',
5     'adresse' => '3 Rue du Paradis',
6     'ville' => 'Marseille');
7
8 if (array_key_exists('nom', $coordonnees))
9 {
10     echo 'La clé "nom" se trouve dans les coordonnées !';
11 }
12
13 if (array_key_exists('pays', $coordonnees))
14 {
15     echo 'La clé "pays" se trouve dans les coordonnées !';
16 }
17
18 ?>
```



# Partie 2 : Bases de PHP

## Les tableaux (arrays) : recherche dans array

- Vérifier si une valeur existe dans l'array : « in\_array »  
<http://php.net/manual/fr/function.in-array.php>
  - Le principe est le même que « array\_key\_exists »
  - mais cette fois on recherche dans les valeurs
  - « in\_array » renvoie :
    - ✓ true si la valeur se trouve dans l'array
    - ✓ false si elle ne s'y trouve pas

# Partie 2 : Bases de PHP

## Les tableaux (arrays) : recherche dans array

- Vérifier si une valeur existe dans l'array : « in\_array »

```
1 <?php
2 $fruits = array ('Banane', 'Pomme', 'Poire', 'Cerise', 'Fraise', 'Framboise');
3
4 if (in_array('Myrtille', $fruits))
5 {
6     echo 'La valeur "Myrtille" se trouve dans les fruits !';
7 }
8
9 if (in_array('Cerise', $fruits))
10 {
11     echo 'La valeur "Cerise" se trouve dans les fruits !';
12 }
13 ?>
```

# Partie 2 : Bases de PHP

## Les tableaux (arrays) : recherche dans array

- Vérifier si une valeur existe dans l'array : « in\_array »

```
1 <?php
2 $fruits = array ('Banane', 'Pomme', 'Poire', 'Cerise', 'Fraise', 'Framboise');
3
4 if (in_array('Myrtille', $fruits))
5 {
6     echo 'La valeur "Myrtille" se trouve dans les fruits !';
7 }
8
9 if (in_array('Cerise', $fruits))
10 {
11     echo 'La valeur "Cerise" se trouve dans les fruits !';
12 }
13 ?>
```

- On ne voit que le message pour la cerise
- « in\_array » a renvoyé :
  - ✓ true pour « Cerise »
  - ✓ false pour « Myrtille »

# Partie 2 : Bases de PHP

## Les tableaux (arrays) : recherche dans array

- Récupérer la clé d'une valeur dans l'array : « array\_search »  
<http://php.net/manual/fr/function.array-search.php>
  - Fonctionne comme « in\_array » : il travaille sur les valeurs d'un array
  - « array\_search » renvoie :
    - ✓ si elle a trouvé la valeur :
      - ❖ « array\_search » renvoie la clé correspondante :
        - le numéro si c'est un array numéroté
        - le nom de la clé si c'est un array associatif
    - ✓ si elle n'a pas trouvé la valeur :
      - ❖ « array\_search » renvoie false

# Partie 2 : Bases de PHP

## Les tableaux (arrays) : recherche dans array

- Récupérer la clé d'une valeur dans l'array : « array\_search »

```
1 <?php
2 $fruits = array ('Banane', 'Pomme', 'Poire', 'Cerise', 'Fraise', 'Framboise');
3
4 $position = array_search('Fraise', $fruits);
5 echo '"Fraise" se trouve en position ' . $position . '<br />';
6
7 $position = array_search('Banane', $fruits);
8 echo '"Banane" se trouve en position ' . $position;
9 ?>
```

Rappel : un array numéroté commence à 0

# Partie 2 : Bases de PHP

## Les tableaux (arrays) : recherche dans array

- Résumé

- ❖ Les tableaux (ou arrays) sont des variables représentées sous forme de tableau.
  - Elles peuvent donc stocker de grandes quantités d'informations.
- ❖ Chaque ligne d'un tableau possède :
  - une clé (qui permet de l'identifier)
  - une valeur.
- ❖ Il existe deux types de tableaux :
  - les tableaux numérotés :
    - chaque ligne est identifiée par une clé numérotée.
    - La numérotation commence à partir de 0 ;
  - les tableaux associatifs :
    - chaque ligne est identifiée par une courte chaîne de texte
- ❖ Pour parcourir un tableau, on peut utiliser :
  - la boucle « for » que l'on connaît déjà
  - mais aussi la boucle « foreach » qui est dédiée aux tableaux.
- ❖ Il existe de nombreuses fonctions permettant de travailler sur des tableaux et notamment d'effectuer des recherches.