PHP

Les tableaux

Dans ce module

- Array simple
- Array complexe
- Foreach sur un tableau
- Travaux sur un tableau
- Array multidimensionnel ou imbriqué
- Fonctions sur les tableaux
- Concaténation de tableaux
- Pratique

Array simple

- Un array (tableau) est un type contenant plusieurs valeurs (données).
- Une variable de type array peut donc contenir plusieurs valeurs.
- Un array se présente sous la forme de crochet [ ].
- Toutes les valeurs peuvent être de n'importe quels types.

```php
[]; // []
[VAL1, VAL2]; // [VAL1, VAL2]
```

Forme raccourci (moderne)

```php
array(); // []
array(VAL1, VAL2); // [VAL1, VAL2]
```

Forme classique

Array simple - Exemple

```php
$arr = ['name', 50, null, '2', [], 7.5, true];

foreach($arr as $a) {
    var_dump($a);
}
```

```
string 'name' (length=4)
50
null
string '2' (length=1)
array (size=0)
  empty
float 7.5
boolean true
```

Array complexe

- En réalité, chaque valeur d'un array en PHP possède une clé.
- La clé est définie explicitement ou générée automatiquement en partant de 0 ou en suivant la dernière clé numérique.
- Le fonctionnement des arrays en PHP est le plus avancé.
  - *Un array PHP équivaut à : array JS + objet JS*
  - *Un array PHP équivaut à : list Python + dictionary Python*

```php
[]; // []
[VAL1, VAL2]; // [0 => VAL1, 1 => VAL2]
['a' => VAL1, 'b' => VAL2]; // ['a' => VAL1, 'b' => VAL2]
[10 => VAL1, 'str' => VAL2]; // [10 => VAL1, 'str' => VAL2]
[
    10 => VAL1,
    'str' => VAL2,
    VAL3,
]; // [10 => VAL1, 'str' => VAL2, 11 => VAL3]
```

Array complexe - Représentation

```
['e', 'b', 'c', 'd', 'a', 'f', 'g']
```

Ordre	0	1	2	3	4	5	6
Key/index	0	1	2	3	4	5	6
Value	e	b	c	d	a	f	g

```
[1=>'e', 0=>'b', 3=>'c', 5=>'d', 2=>'a', 4=>'f', 6=>'g']
```

Ordre	0	1	2	3	4	5	6
Key/index	1	0	3	5	2	4	6
Value	e	b	c	d	a	f	g

Tableaux

Array complexe - Exemple

```php
$utf8 = [
    41 => 'A',
    42 => 'B',
    61 => 'a',
    62 => 'b',
];
```

```php
echo $utf8[42]; // B
echo $utf8[61]; // a
```

```php
$google = [
    'name' => 'Google',
    'created_at' => 1998,
    'used' => 94.2,
];
```

```php
echo $google['name']; // Google
echo $google['used']; // 94.2
```

Foreach sur un tableau

- On peut utiliser un foreach pour itérer sur chaque élément du tableau.
- On peut alors utiliser 2 formes :
  - *On ne récupère que la valeur des éléments du tableau*
  - *On récupère la clé et la valeur de chaque élément du tableau*

```php
foreach(ARRAY as $VALUE) {
    // CODE
}
```

Foreach avec uniquement la valeur.

```php
foreach(ARRAY as $KEY => $VALUE) {
    // CODE
}
```

Foreach avec la clé et la valeur.

Foreach sur un tableau - Exemple

```php
$screen = ['720p', '1080p', '1440p', '2K', '4K'];

foreach ($screen as $s) {
    echo $s.'<br>';
}
```

720p
1080p
1440p
2K
4K

```php
$google = [
    'name' => 'Google',
    'created_at' => 1998,
    'used' => 94.2,
];

foreach ($google as $key => $data) {
    echo $key.' : '.$data.'<br>';
}
```

name : Google
created_at : 1998
used : 94.2

Travaux sur un tableau

- Comme toutes les valeurs d'un tableau PHP possèdent une clé, on peut réaliser les actions suivantes :
  - *Ajouter un élément clé/valeur à un tableau existant*

```php
$arr = [];
$arr[KEY] = VALUE;
// [KEY => VALUE];

$cat = ['type' => 'chat'];
$cat['race'] = 'norvégien';
// ['type' => 'chat', 'race' => 'norvégien']
```

  - *Modifier une valeur à partir d'une clé connue*

```php
$cat['race'] = 'main coon';
// ['type' => 'chat', 'race' => 'main coon']
```

  - *Supprimer un élément clé/valeur*

```php
unset($cat['type']);
// ['race' => 'main coon']
```

Travaux sur un tableau

- Comme toutes les valeurs d'un tableau PHP possèdent une clé, on peut réaliser les actions suivantes :
  - *Ajouter, modifier ou supprimer des éléments clé/valeur depuis un foreach :*

```php
$cat = [
    'type' => 'chat',
    'race' => 'norvégien',
    'origin_type' => 'natural breed',
    'origin' => 'northern europe',
];
$keyToChange = ['race' => 'breed'];

foreach ($cat as $key => $value) {
    // Uppercase first char of all values
    $value = ucfirst($value);
    // Replace the space char by a dash
    $value = str_replace(' ', '-', $value);
    // Update value
    $cat[$key] = $value;

    // Change key
    if (array_key_exists($key, $keyToChange)) {
        $newKey = $keyToChange[$key]; // Get new key name
        $cat[$newKey] = $cat[$key]; // Add new element key/value
        unset($cat[$key]); // Remove old key/value
    }
}
```

```
array (size=4)
  'type' => 'Chat'
  'origin_type' => 'Natural-breed'
  'origin' => 'Northern-europe'
  'breed' => 'Norvégien'
```

Tableaux

Array multidimensionnel ou imbriqué

- Les valeurs d'un tableau peut être de n'importe quels type donc on peut insérer un tableau dans un tableau.
- On parlera alors de tableaux multidimensionnel. Le nombre de dimension est théoriquement infini.

```
[
    KEY => [
        KEY => VAL,
    ],
    KEY => [
        KEY => [
            KEY => [
                KEY => VAL,
                KEY => VAL,
            ],
        ],
    ],
]
```

Tableaux

Array multidimensionnel - Exemple

```php
$searchEngine = [
    'google' => [
        'name' => 'Google',
        'owner' => 'Google LLC',
        'country' => 'usa',
        'created_at' => 1998,
        'used' => 94.2,
    ],
    'bing' => [
        'name' => 'Bing',
        'owner' => 'Microsoft Corporation',
        'country' => 'usa',
        'created_at' => 2008,
        'used' => 2.9,
    ],
    'yahoo' => [
        'name' => 'Yahoo!',
        'owner' => 'Verizon Communications Inc.',
        'country' => 'usa',
        'created_at' => 1994,
        'used' => 1.5,
    ],
    'qwant' => [
        'name' => 'Qwant',
        'owner' => 'Qwant SAS',
        'country' => 'french',
        'created_at' => 2013,
        'used' => 0.7,
    ],
];
```

```php
// Display value
echo $searchEngine['google']['owner']; // Google LLC
// Update value
$searchEngine['qwant']['name'] = 'QWANT';
// Add key/value
$searchEngine['qwant']['open'] = true;
```

```php
foreach($searchEngine as $seKey => $se) {
    $title = isset($se['name']) ? $se['name'] : $seKey;

    echo '<h4>'.strtoupper($title).'</h4>';
    echo '<ul>';
    foreach($se as $key => $value) {
        echo '<li>'.$key.': '.$value.'</li>';
    }
    echo '</ul>';
}
```

GOOGLE

- name: Google
- owner: Google LLC
- country: usa
- created_at: 1998
- used: 94.2

...

Fonctions sur les tableaux

```php
// Trie par les valeurs avec nombre
$arr = [1, 55.4, 78, -3, 19];
sort($arr); // [-3, 1, 19, 55.4, 78]
```

```php
// Trie par les valeurs avec string
$arr = ['franboise', 'fraise', 'coco'];
sort($arr); // ['coco', 'fraise', 'franboise']
```

```php
// Trie par les clés
$arr = [1 => 'coco', 0 => 'fraise'];
ksort($arr); // [0 => 'fraise', 1 => 'coco']
```

```php
// Inverse l'ordre (renverse)
$arr = ['ppt', 7, ['bleu', 'red']];
$arr = array_reverse($arr);
// [['bleu', 'red'], 7, 'ppt']
```

```php
// La clé existe ?
$arr = ['name' => 'Google', 'created_at' => 1998];
array_key_exists('created_at', $arr); // true
```

```php
// La valeur est dans le tableau ?
$arr = ['name' => 'Google'];
in_array('Google LLS', $arr); // false
```

```php
// Nombre d'élément ?
$arr = [1, 55.4, 78, -3, 19];
$nb = count($arr); // 5
```

```php
// Je veux seulement les valeurs !
$arr = ['name' => 'Google', 'created_at' => 1998];
$values = array_values($arr);
// [0 => 'Google', 1 => 1998]
```

Fonctions sur les tableaux

```php
// Avec séparetur : Array => String
$arr = ['Je', 'suis', 'là'];
$str = implode('-', $arr); // Je-suis-là
```

```php
// Sans séparetur : Array => String
$arr = ['Je', 'suis', 'là'];
$str = implode('', $arr); // Jesuislà
```

```php
// Avec séparetur : String => Array
$str = 'blabla@gmail.com';
$arr = explode('@', $str); // [0 => 'blabla', 1 => 'gmail.com']
```

```php
// Sans spérateur : String => Array
$str = 'Hello';
$arr = str_split($str); // ['H', 'e', 'l', 'l', 'o']
$arr = str_split($str, 2); // ['He', 'll', 'o']
```

➔ Toutes les fonctions sur : https://www.php.net/manual/fr/ref.array.php

Concaténation de tableaux

```php
$a = [
    "a" => "pomme",
    "b" => "banane"
];

$b = [
    "a" =>"poire",
    "b" => "fraise",
    "c" => "cerise"
];

var_dump($a + $b);
var_dump($b + $a);
var_dump(array_merge($a, $b));
```

```
array (size=3)
  'a' => string 'pomme' (length=5)
  'b' => string 'banane' (length=6)
  'c' => string 'cerise' (length=6)

array (size=3)
  'a' => string 'poire' (length=5)
  'b' => string 'fraise' (length=6)
  'c' => string 'cerise' (length=6)

array (size=3)
  'a' => string 'poire' (length=5)
  'b' => string 'fraise' (length=6)
  'c' => string 'cerise' (length=6)
```

Concaténation de tableaux

```php
$a = [
    0 => "pomme",
    1 => "banane"
];

$b = [
    0 => "poire",
    1 => "fraise",
    2 => "cerise"
];

var_dump($b + $a);
var_dump(array_merge($a, $b));
```

```
array (size=3)
  0 => string 'poire' (length=5)
  1 => string 'fraise' (length=6)
  2 => string 'cerise' (length=6)

array (size=4)
  0 => string 'pomme' (length=5)
  1 => string 'banane' (length=6)
  2 => string 'poire' (length=5)
  3 => string 'fraise' (length=6)
  4 => string 'cerise' (length=6)
```

Pratique

1. Testez vous-même les exemples du cours et vérifier les résultats. Utiliser « var_dump » pour vérifier les résultats.
2. Testez les fonctions vues dans ce cours et changez les tableaux des exemples puis observez les résultats. Le but étant de mieux comprendre les fonctions essentielles à la manipulation des tableaux.

3. Créez un programme de validation de mot de passe (une variable) suivant les règles :
   - Au moins 8 charactères
   - Au moins 2 chiffres
   - Au moins 1 lettres minuscules et majuscules
   - Au moins 2 caractères spéciaux parmi : * ^ % # - + ? ! , ; : = @

4. Créez un programme de validation d'adresse email (une variable) suivant les règles :

   - Email de forme: **STRING @ STRING . STRING** (sans espace)
   - La 1ère et 2ème STRING accepte uniquement le caractère spécial « point »
   - La 1ère et 2ème STRING doit obligatoirement contenir des lettres et peut contenir des chiffres
   - La 3ème STRING ne peut contenir que des lettres a-z (pas de chiffre, pas de caractère spécial)

PHP

Les tableaux

Fin du module