

# HTML / CSS / PHP

## Todolist - CHAPITRE 2

# Intégration de PHP

## Objectif

Créer les différentes items de la liste des tâches via un tableau en PHP, puis utilisez une boucle pour l’affichage des éléments du tableau.

Conditionner la checkbox du second item à être coché.

## Points abordés

- Les variables, les types de données, les opérations
- Découvrir les structures de contrôles en PHP
- Découvrir les tableaux en PHP

## Todo List

### Gestionnaire de tâches

#### Ajouter une tâche

ou

☐

Passer le balai

☒

Saluer le boss

☐

Couper l'ordi

# PHP

## Documentations

### Les variables

<https://www.php.net/manual/fr/language.variables.basics.php>

### Les types de données

<https://www.php.net/manual/fr/language.types.php>

### Les opérateurs

<https://www.php.net/manual/fr/language.operators.php>

# PHP

## Documentations

### Les tableaux en PHP

En PHP, un tableau est une structure de données qui permet de stocker et d'organiser un ensemble de valeurs sous une seule variable. Ces valeurs peuvent être de différents types, notamment des nombres, des chaînes de caractères, d'autres tableaux, des objets, etc. Les tableaux sont extrêmement flexibles et polyvalents, ce qui en fait l'une des structures de données les plus couramment utilisées en programmation PHP.

<https://www.php.net/manual/fr/intro.array.php>

# PHP

## Documentations

Les tableaux en PHP peuvent être de deux types principaux

### Tableaux indexés :

Dans un tableau indexé, chaque élément est associé à un indice numérique, qui commence généralement à partir de zéro. Les indices sont attribués automatiquement si vous ne les spécifiez pas explicitement.

```
$fruits = array('Pomme', 'Banane', 'Fraise');
```

### Tableaux associatifs :

Dans un tableau associatif, chaque élément est associé à une clé (un nom) plutôt qu'à un indice numérique. Vous spécifiez la clé pour chaque élément.

```
$personne = array('nom' => 'Smith', 'prénom' => 'John', 'âge' => 30);
```

Pour accéder aux éléments d'un tableau, vous utilisez l'indice ou la clé associée à l'élément. Par exemple, pour accéder à la première valeur du tableau indexé des fruits, vous utiliseriez `$fruits[0]`, tandis que pour accéder à l'âge dans le tableau associatif de la personne, vous utiliseriez `$personne['âge']`.

# PHP

## Documentations

### Il est important de considérer les tableaux multidimensionnels

Les tableaux multidimensionnels en PHP sont des tableaux qui contiennent d'autres tableaux en tant qu'éléments. Ils sont utilisés pour stocker des données de manière hiérarchique, souvent sous forme de tableaux imbriqués à deux dimensions (tableaux dans un tableau). Les tableaux multidimensionnels permettent de représenter des structures de données complexes, telles que des matrices, des grilles, des enregistrements de base de données, etc.

```
$personnes = array(  
    'personne1' => array('nom' => 'Smith', 'âge' => 30),  
    'personne2' => array('nom' => 'Doe', 'âge' => 25)  
);
```

```
var_dump($personnes['personne1']);  
  
// array(2) {  
//     ["nom"]=>  
//     string(5) "Smith"  
//     ["âge"]=>  
//     int(30)  
// }
```

# PHP

## Documentations

### Les structures de contrôle

Les structures de contrôle en programmation sont des éléments fondamentaux qui permettent de gérer le flux d'exécution d'un programme. Elles permettent de prendre des décisions, de répéter des actions, d'exécuter des blocs de code conditionnellement, et de contrôler l'ordre dans lequel les instructions sont exécutées. Les structures de contrôle sont essentielles pour créer des programmes qui réagissent dynamiquement aux données et aux situations.

**<https://www.php.net/manual/fr/language.control-structures.php>**

# PHP

## Documentations

### 1) Structure de séquence :

Les instructions sont exécutées dans l'ordre séquentiel dans lequel elles sont écrites, de haut en bas. C'est le flux d'exécution standard d'un programme, où chaque instruction est exécutée après la précédente.

### 2) Structure de décision (ou conditionnelle) :

Les structures de décision permettent d'exécuter des blocs de code en fonction d'une condition. Les principales structures de décision sont :

- `if` : Exécute un bloc de code si une condition est vraie.
- `else` : Exécute un bloc de code alternatif si la condition de `if` est fausse.
- `elseif` (ou `else if`) : Permet de vérifier plusieurs conditions en séquence.
- `switch` : Permet de choisir parmi plusieurs blocs de code en fonction de la valeur d'une expression.



# PHP

## Documentations

**3) Structure de répétition (ou boucle) :** Les structures de répétition permettent d'exécuter un bloc de code plusieurs fois, en fonction d'une condition. Les principales structures de répétition sont :

- `for` : Exécute un bloc de code un nombre spécifié de fois.
- `while` : Exécute un bloc de code tant qu'une condition est vraie.
- `do...while` : Exécute un bloc de code au moins une fois, puis tant qu'une condition est vraie.
- `foreach` : Utilisé pour parcourir les éléments d'une liste ou d'un tableau.

**4) Structure de saut :** Les structures de saut permettent de transférer le contrôle de l'exécution du programme à un endroit spécifique du code. Les principales structures de saut sont :

- `break` : Termine l'exécution d'une boucle ou d'un commutateur.
- `continue` : Passe à l'itération suivante d'une boucle.
- `return` : Termine l'exécution de la fonction actuelle et renvoie une valeur.

# PHP

## Documentations

### Focus sur foreach

**foreach** est une boucle de type itératif qui parcourt tous les éléments d'une structure de données (tableau ou objet) et exécute un bloc de code pour chaque élément. À chaque itération de la boucle, une variable spéciale prend la valeur de l'élément en cours d'itération, ce qui permet d'accéder à cet élément dans le bloc de code.

```
$personne = array(  
    'nom' => 'Smith',  
    'prénom' => 'John',  
    'âge' => 30,  
    'ville' => 'Paris'  
);  
  
foreach ($personne as $cle => $valeur) {  
    echo "Clé : $cle, Valeur : $valeur <br>";  
}
```

# PHP

## Documentations

### Focus sur if

**if** est une instruction conditionnelle en PHP qui permet d'exécuter un bloc de code seulement si une expression ou une condition est évaluée comme vraie (true). Si la condition est vraie, le code à l'intérieur du bloc if est exécuté ; sinon, il est ignoré.

```
$age = 25;

if ($age < 18) {
    echo "Vous êtes mineur.";
} else {
    echo "Vous êtes majeur.";
}
```

# Exercice

Créer les différentes items de la liste des tâches via un **tableau** en PHP, puis utilisez une **boucle** pour l’affichage des éléments du tableau.

**Conditionner** la checkbox du second item à être coché.

Temps : 2H

## Todo List

### Gestionnaire de tâches

#### Ajouter une tâche

ou

☐

Passer le balai

☒

Saluer le boss

☐

Couper l'ordi

# index.php

## Tableaux / boucle

Création du tableau  
multidimensionnel

Parcours du tableau ->  
[https://www.php.net/  
manual/fr/control-  
structures.foreach.php](https://www.php.net/manual/fr/control-structures.foreach.php)

```
<?php
$tasks = array(
    array('task' => "Passer le balai", 'id' => 1, 'status' => "wip"),
    array('task' => "Saluer le boss", 'id' => 2, 'status' => "finish"),
    array('task' => "Couper l'ordi", 'id' => 1, 'status' => "wip")
)
?>
<!DOCTYPE html>
<html lang="fr">
> <head> ...
    </head>
    <body>
        <h1>Todo List</h1>
        <h2>Gestionnaire de tâches</h2>
> <form class="form-add" method="post"> ...
    </form>

    <ul class="list-todo">
        <?php if($tasks) : ?>
            <?php foreach($tasks as $key => $task) : ?>
                <li class="<?php echo $task['status'] == "finish" ? "ok": ""; ?>">
                    <input <?php echo $task['status'] == "finish" ? "checked": ""; ?> type="checkbox" name="status">
                    <span><?php echo $task['task']; ?></span>
                    <a href="?mode=delete&id=<?php echo $task['id']; ?>" class="btdelete"></a>
                </li>
            <?php endforeach; ?>
        <?php endif; ?>
    </ul>
</body>
</html>
```

# index.php

## Conditions

### Focus dans la boucle

```
<?php
$tasks = array(
    array('task' => "Passer le balai", 'id' => 1, 'status' => "wip"),
    array('task' => "Saluer le boss", 'id' => 2, 'status' => "finish"),
    array('task' => "Couper l'ordi", 'id' => 1, 'status' => "wip")
)
?>
<!DOCTYPE html>
<html lang="fr">
> <head> ...
</head>
<body>
    <h1>Todo List</h1>
    <h2>Gestionnaire de tâches</h2>
> <form class="form-add" method="post"> ...
</form>

    <ul class="list-todo">
        <?php if($tasks) : ?>
            <?php foreach($tasks as $key => $task) : ?>
                <li class="<?php echo $task['status'] == "finish" ? "ok": ""; ?>">
                    <input <?php echo $task['status'] == "finish" ? "checked": ""; ?> type="checkbox" name="status">
                    <span><?php echo $task['task']; ?></span>
                    <a href="?mode=delete&id=<?php echo $task['id']; ?>" class="btdelete"></a>
                </li>
            <?php endforeach; ?>
        <?php endif; ?>
    </ul>
</body>
</html>
```

# index.php

## Conditions

Gestion de la checkbox

Via des conditions ternaires -> <https://www.php.net/manual/fr/language.operators.comparison.php#language.operators.comparison.ternary>

```
<li class="<?php echo $task['status'] == "finish" ? "ok": ""; ?>">
  <input
    <?php echo $task['status'] == "finish" ? "checked": ""; ?>
    type="checkbox"
    name="status"
  >
  <span>
    <?php echo $task['task']; ?>
  </span>
  <a
    href="?mode=delete&id=<?php echo $task['id']; ?>"
    class="btdelete">
  </a>
</li>
```