PHP

Passez à la vitesse supérieure avec PHP

Plan du cours

- Introduction à PHP
  - C’est quoi ?
  - Utilisation dans le développement web
- Architecture Client / Server
- Communication entre Client et Serveur
- Fonctionnement de site web statique et dynamique
- Environnement de développement
  - Téléchargement et installation d’un serveur (XAMPP, WAMP, MAMP,...)
- Premier pas avec PHP
  - Structure d’une page PHP
  - Affichage de texte
  - Commentaires

Plan du cours

- Variables et types de données
  - Déclaration de variable
  - Types de données de base
  - Opération sur les données
- Les structures conditionnelles
  - if, else, elseif, switch
- Les structure itératives
  - for, while, do while
- Inclusions de fichiers
  - include
  - require
- Les fonctions

Plan du cours

- Les tableaux
  - Tableaux indicés
  - Tableau associatifs
  - Parcours et affichage des éléments d’un tableau
- Formulaires HTML et Traitement PHP
  - Création des formulaire HTML
  - Soumission de formulaire et traitement des données
  - Validation des données côté serveur
- Interaction avec une Base de données CRUD
  - Requête d’insertion
  - Requête de lecture
  - Requête de mise à jour
  - Requête de de suppression

Plan du cours

- Sessions et Cookies
  - Utilisation des sessions pour maintenir l'état des utilisateurs
  - Gestion des cookies
- Gestion de fichiers
  - Lecture
  - Ecriture
- Hébergement d’un site

Les boucles

Les boucles sont pratiques si vous souhaitez exécuter le même code plusieurs fois avec une valeur différente à chaque fois. C'est souvent le cas lorsque vous travaillez avec des tableaux.

PHP prend en charge différents types de boucles :

- **for** parcourt un bloc de code un certain nombre de fois
- **while** parcourt un bloc de code tant qu'une condition spécifiée est vraie
- **do while** parcourt également un bloc de code tant qu'une condition spécifiée est vraie.

La boucle for

```php
for(initialisation; condition; incrémentation){
    // instructions à exécuter
}

// exemple
for($i = 0; $i <= 5; $i++){
    echo "<p>".$i."</p>";
}
```

La boucle while

```php
$i = 0; // initialisation
while(condition){
    // instructions à exécuter
    // incrémentation
}

// exemple:
$i = 0;
while($i <= 0){
    echo "<p>".$i."</p>";
    $i++;
}
```

La boucle do while

```php
$i = 0; // initialisation
do{
    // instructions à exécuter
}while(condition);

// exemple
$i = 0;
do{
    echo "<p>".$i."</p>";
    $i++;
}while($i <= 5);
```

Inclusion de fichier

En PHP, vous pouvez inclure le contenu d'un fichier dans un autre à l'aide de plusieurs méthodes d'inclusion de fichiers. Cela peut être utile pour réutiliser du code commun, organiser votre code en modules ou séparer la logique de présentation.

Voici les principales méthodes d'inclusion de fichiers en PHP :

include, require, include_once et require_once

Inclusion de fichier

Include :

La fonction include permet d'inclure le contenu d'un fichier dans un autre. Si le fichier à inclure n'existe pas, une simple alerte sera émise, et le script continuera son exécution.

```
include 'chemin/vers/fichier.php';
```

Inclusion de fichier

Require :

La fonction require est similaire à include, mais si le fichier inclus n'est pas trouvé, une erreur fatale sera générée et l'exécution du script sera arrêtée.

```
require 'chemin/vers/fichier.php';
```

Inclusion de fichier

Include_once et **require_once** :

Ces deux fonctions sont similaires à include et require, mais elles vérifient d'abord si le fichier a déjà été inclus dans le script. Cela évite d'inclure plusieurs fois le même fichier, ce qui pourrait causer des erreurs comme : Définitions multiples de fonctions ou de classes, Variables redéfinies...

```php
include_once 'chemin/vers/fichier.php';
require_once 'chemin/vers/fichier.php';
```

Les fonctions

Une fonction est un bloc de code autonome et réutilisable dans un programme informatique. Elle accomplit une tâche spécifique ou exécute une série d'instructions définies. Les fonctions sont utilisées pour organiser le code en unités logiques, permettant ainsi de simplifier la complexité, d'améliorer la lisibilité et de favoriser la réutilisation du code.

Les fonctions

Dans la plupart des langages de programmation, y compris PHP, une fonction est caractérisée par les éléments suivants :

1. **Nom** : Chaque fonction a un nom unique qui la distingue des autres. Le nom d'une fonction est utilisé pour l'appeler et exécuter le code qu'elle contient.
2. **Paramètres** : Les fonctions peuvent accepter des paramètres, qui sont des valeurs transmises à la fonction lors de son appel. Ces paramètres fournissent des informations à la fonction pour effectuer son travail.
3. **Corps** : Le corps de la fonction contient les instructions et le code à exécuter lorsque la fonction est appelée. C'est ici que vous définissez ce que la fonction doit accomplir.
4. **Valeur de retour** : Une fonction peut renvoyer une valeur après avoir effectué ses calculs ou traitements. Cette valeur peut être utilisée dans d'autres parties du code.

Les fonctions

Déclaration de fonction :

Pour déclarer une fonction, utilisez le mot-clé function, suivi du nom de la fonction et de ses éventuels paramètres entre parenthèses. Le bloc de code à exécuter est entouré de accolades {}.

```
function nomDeLaFonction($parametre1, $parametre2) {
        // Code à exécuter
        // ...
}
```

Les fonctions

Appel de fonction :

Pour appeler une fonction, utilisez simplement son nom suivi des valeurs pour les paramètres entre parenthèses si la fonction en prend.

```
nomDeLaFonction($valeur1, $valeur2);
```

Les fonctions

Retour de valeurs :
Une fonction peut également renvoyer une valeur à partir de son résultat. Utilisez le mot-clé return pour spécifier la valeur à renvoyer.

```
function addition($a, $b) {
	$resultat = $a + $b;
	return $resultat;
}

$resultatAddition = addition(5, 3); // $resultatAddition contiendra 8
```

Les tableaux

En programmation, un tableau est une structure de données qui permet de stocker plusieurs valeurs du même type ou de types différents sous un même nom. Ils sont utilisés pour organiser et manipuler des données de manière structurée.

Il existe plusieurs types de tableaux en PHP, notamment les tableaux indicés (indexés) et les tableaux associatifs.

Les tableaux

Tableaux indicés (indexés) :

Les tableaux indexés sont des collections d'éléments indexés par des nombres entiers, commençant par 0.

```php
$fruits = ["pomme", "banane", "orange"];

echo $fruits[0]; // Affiche "pomme"
echo $fruits[1]; // Affiche "banane"
echo $fruits[2]; // Affiche "orange"
```

Les tableaux

Tableaux associatifs :

Les tableaux associatifs sont des collections d'éléments indexés par des clés (noms) au lieu de nombres.

```
$personne = array(
        "nom" => "Doe",
        "prénom" => "John",
        "âge" => 30
);

// Accès aux éléments par clé
echo $personne["nom"]; // Affiche "Doe"
echo $personne["âge"]; // Affiche 30
```

Les tableaux

PHP propose de nombreuses fonctions intégrées pour manipuler et travailler avec des tableaux. Voici quelques-unes des méthodes couramment utilisées pour manipuler les tableaux en PHP :

Ajouter des éléments :

- `array_push($array, $element); // Ajoute un ou plusieurs éléments à la fin du tableau`
- `$array[] = $element; // Ajoute un élément à la fin du tableau`

Supprimer des éléments :

- `array_pop($array); // Supprime le dernier élément du tableau et le renvoie`
- `array_shift($array); // Supprime le premier élément du tableau et le renvoie`

Les tableaux

Compter les éléments :

- `count($array); // Retourne le nombre d'éléments dans le tableau`
- `sizeof($array); // Identique à count($array)`

Vérifier si une clé existe :

- `array_key_exists($key, $array); // Vérifie si une clé existe dans le tableau`

Recherche d'éléments :

- `in_array($element, $array); // Vérifie si un élément existe dans le tableau`
- `array_search($element, $array); // Recherche un élément dans le tableau et renvoie`

Parcourir un tableau :

- `foreach ($array as $key => $value) // Parcourt tous les éléments du tableau`
- `foreach ($array as $value) // Parcourt tous les éléments du tableau`

Les tableaux

Tri de tableaux :

- `sort($array); // Trie les éléments d'un tableau en ordre croissant`
- `rsort($array); // Trie les éléments d'un tableau en ordre décroissant`

Filtrage de tableaux :

- `array_filter($array, $callback); // Filtre les éléments d'un tableau en utilisant une fonction de rappel.`
- `array_map($callback, $array); // Applique une fonction à chaque élément d'un tableau et renvoie un nouveau tableau.`