PHP

Développement web avec PHP

FORMATEUR : MAHAMANE SALISSOU YAHAYA : ysalissou@gmail.com

(TechnoLAB -ISTA)
Master I GL

Décembre 2024

A propos du formateur

- Mahamane Salissou YAHAYA
- Email : ysalissou@gmail.com
- Tél : +223 74 46 63 17

- Consultant en système d'information
- Résumé des compétences:
  - Développement JAVA/J2EE, PHP, DOTNET, ANDROID
  - Administrateur base de données
  - Mise en place et audit de système d'information de gestion
  - Conduite de projet informatique
  - Formateur vacataire

- Actuellement directeur technique chez SITAN INFORMATIQUE.
- Mes références :

  Linkedin : https://www.linkedin.com/in/salissou-yahaya-2a1b7071/

A propos des étudiants

- **Nom**
- **Expériences en développement**
- **Objectif de ce cours**

Horaires de la formation

- **Durée : 24 heures**
- **Horaire : de 17h00 à 20h00**
- **15 minutes de pause au milieu de chaque séance**

Organisation de l'enseignement

- **Objectifs**
  - Apprenez le PHP sur sa version la plus récente avec la version 8
  - Familiarisez vous avec les fonctionnalitées natives du langage
  - Apprenez les syntaxes modernes du langage
  - Maitrisez la programmation orientée objet avec les classes, les traits et les interfaces
  - Maitrisez toutes les bases du langage avec les types, les tableaux et les fonctions
  - Apprenez à mettre vos applications en production avec Nginx, PHP-FPM et HTTPS
  - Découvrez comment utiliser MySQL dans vos applications
  - Découvrez Composer et l'autoloading
- **Prérequis**
  - Des connaissance en HTML & CSS sont nécessaires
  - Et … du courage comme toujours !
- **Évaluations**
  - Un QCM sur les notions importantes et projet de fin de module
  - Il est important de venir à tous les TP
  - Les notes sont individuelles

Projet réalisé pendant la formation

Nous utiliserons l'ensemble des fonctionnalités de PHP pour mettre en place un projet complet dans lequel nous créerons un blog.

Nous utiliserons également les éléments suivants :

MySQL et PDO

Toutes les applications backend, comme celles créées avec PHP, nécessitent d'utiliser une base de données. En PHP, on utilise le plus souvent des bases de données SQL. La base de données la plus utilisée est MySQL. Durant la formation vous apprendrez toutes les bases du SQL et vous apprendrez à faire communiquer vos applications PHP avec MySQL.

Nginx

Pour que vos utilisateurs puissent utiliser votre application PHP, il faut pouvoir leur délivrer via un serveur Web. Le plus utilisé aujourd'hui et le plus performant est NGINX. C'est pourquoi durant la formation nous vous montrerons comment mettre en place une architecture moderne avec Nginx et PHP-FPM pour donner à vos utilisateurs la meilleure expérience possible avec des vitesses de chargement extrêmement rapides.

Programmation orientée objet

La programmation orientée objet est permise en PHP grâce à l'utilisation d'un système de classe. Les classes permettent de regrouper des fonctionnalités pour permettre une meilleure lisibilité de votre code.Dans la formation vous apprendrez toutes les bases de la POO et comment la mettre en pratique.

Description

Le PHP est le langage backend le plus utilisé au monde pour la réalisation d'applications Web.

Le langage s'utilise en complément de HTML, CSS ainsi que de JavaScript.

Le rôle du PHP est de vous permettre de créer toute la partie backend de vos applications, c'est grâce à lui que vous allez pouvoir gérer l'authentification de vos utilisateurs et sauvegarder des données dans une base de données.

80% des sites Web actuels ont été réalisés avec PHP.

C'est un langage facile d'accès mais avec beaucoup de profondeur qui prendra des années à maîtriser parfaitement.

Vous serez à jour de toutes les nouveautés qui se sont ajoutées au fil des années (PHP 5, PHP 6, PHP 7 et PHP 8) et qui font du PHP un langage riche et performant.

PHP est un langage ancien mais qui évolue avec son temps. Vous apprendrez toutes les syntaxes récentes du langage apparues avec la version 8.

Le PHP permet également de faire de la programmation orientée objet avec un système de classes poussé qui vous permettra de réaliser des applications complexes.

Apprendre le langage PHP vous sera très utile pour l'apprentissage de frameworks basés sur PHP comme Symfony ou encore Laravel.

PHP est beaucoup utilisé en entreprise et très souvent il est même obligatoire de le maitriser. Le connaître vous donnera de l'assurance dans votre métier de développeur Web.

Plan du module

Pour débuté

- Chapitre 1 : Introduction au développement Web
- Chapitre 2 : Bases de PHP
- Chapitre 3 : Manipulation des données
- Chapitre 4 : Bases de données avec PHP
- Chapitre 5 : Gestion des sessions et cookies

Thèmes avancés

- Chapitre 6 : PHP avancé
- Chapitre 7 : Travail avec les frameworks PHP
- Chapitre 8 : Projet final

Ressources Recommandées

- Documentation PHP officielle : php.net
- Laravel : laravel.com
- Tutoriels vidéos : FreeCodeCamp, OpenClassrooms, YouTube
- Livres : PHP & MySQL: Server-side Web Development de Jon Duckett

LET' S

GO

PHP

Introduction au développement Web

Comprendre le Web

Dans cette partie, nous allons explorer les fondamentaux du fonctionnement du web. Cela inclut les protocoles, les rôles des serveurs et des clients, ainsi que les types de contenus échangés

Comprendre le Web

Le fonctionnement de base du web

Le web repose sur l'interaction entre un **client** (par exemple, un navigateur) et un **serveur** (par exemple, un serveur Apache). Cette interaction se fait via le **protocole** HTTP (Hypertext Transfer Protocol).

Étapes d'une interaction typique :

1. **Requête HTTP** :
   - Le client envoie une requête au serveur pour demander une ressource (une page web, une image, etc.).
   - Exemple de requête : GET /index.html HTTP/1.1.
2. **Réponse HTTP** :
   - Le serveur traite la requête et retourne une réponse contenant les données demandées (HTML, JSON, etc.) et un code de statut (par exemple, 200 OK pour indiquer que la requête a été satisfaite).
3. **Rendu** :
   - Le navigateur interprète la réponse (souvent en HTML) et affiche la page web.

Types de requêtes HTTP :

- **GET** : Demande de récupérer une ressource (par exemple, afficher une page web).
- **POST** : Envoi de données au serveur (par exemple, soumettre un formulaire).
- **PUT** : Mise à jour d'une ressource existante.
- **DELETE** : Suppression d'une ressource.

Comprendre le Web

Rôles des Clients et des Serveurs

- **Client :**
  - Logiciel qui fait une demande de ressource au serveur.
  - Les clients courants incluent :
    - Les navigateurs web (Chrome, Firefox).
    - Les applications mobiles et de bureau.

- **Serveur :**
  - Logiciel (ou matériel) qui reçoit, traite et répond aux demandes du client.
  - Exemples de serveurs :
    - Serveur web : Apache, Nginx.
    - Serveur d'application : Node.js, PHP.
    - Serveur de base de données : MySQL, PostgreSQL.

- **Schéma d'Interaction Client-Serveur :**

```
Client (navigateur) -----> Requête HTTP -----> Serveur Web
        <----- Réponse HTTP (HTML, CSS, JS) <-----
```

Comprendre le Web

Types de contenus échangés

Le contenu échangé entre le client et le serveur peut inclure :

- **HTML** : Structure de la page web.
- **CSS** : Apparence (couleurs, polices).
- **JavaScript** : Interaction et dynamisme.
- **Images** : Fichiers (PNG, JPEG, SVG).
- **Données** : JSON, XML pour les API.

HTTP et HTTPS

- **HTTP (HyperText Transfer Protocol) :**
  - Protocol de communication non sécurisé.
  - Les données circulent en clair.

- **HTTPS (HTTP Secure) :**
  - HTTP + chiffrement SSL/TLS.
  - Sécurise la communication et protège contre les attaques de type écoute.

- **Exemple de différence** :

```
HTTP : http://example.com
HTTPS : https://example.com
```

Comprendre le Web

Outils et pratiques

- **Navigateur Web :**
  - Les navigateurs modernes comme **Chrome**, Firefox et **Edge** possèdent des outils intégrés pour inspecter les pages web.
    - Appuyez sur **F12** ou faites un **clic droit > Inspecter**.
- **Serveurs locaux :**
  - Pour apprendre PHP, vous pouvez utiliser un serveur local :
  - XAMPP ou WAMP : Fournissent PHP, MySQL et Apache/Nginx.

Exemples pratiques

1. Premier site HTML

Créez un fichier nommé **index.html** :

```
<!DOCTYPE html>
<html>
<head>
    <title>Mon Premier Site</title>
</head>
<body>
    <h1>Bienvenue sur mon site</h1>
    <p>Ce site est généré grâce à HTML.</p>
</body>
</html>
```

Exemples pratiques

2. Ajout de PHP

Créez un fichier nommé **index.php** dans un serveur local :

```
<?php
echo "<h1>Bienvenue sur mon site en PHP</h1>";
?>
```

3. Test des requêtes

Utilisez un outil comme Postman ou curl pour envoyer des requêtes à un serveur local.

Exemple avec curl dans le terminal

```
curl –X GET http://localhost/index.php
```

Comprendre le Web

Exercice

1. Expliquez en vos mots ce qu'est un protocole HTTP.

2. Identifiez les composants suivants dans une URL :

```
https://www.example.com:8080/page?param=valeur#section
```

- **Protocole** : https
- **Nom de domaine** : www.example.com
- **Port** : 8080
- **Chemin** : /page
- **Paramètres** : param=valeur
- **Fragment** : #section

1. Créez un fichier PHP qui affiche la date et l'heure actuelles :

```php
<?php
echo "La date actuelle est : " . date("d/m/Y H:i:s");
?>
```

Comprendre le Web

Quiz rapide

1. Que signifie HTTP ?

    a) HyperText Transfer Protocol
    b) HyperText Translation Protocol
    c) Hyper Transfer Text Protocol

2. Quelle est la principale différence entre HTTP et HTTPS ?

    a) HTTPS est plus rapide.
    b) HTTPS est chiffré.
    c) HTTPS utilise un autre langage.

Les outils essentiels

Pour développer en PHP et créer des applications web, vous aurez besoin de plusieurs outils. Ces outils vous permettront d'écrire, tester et déployer votre code efficacement.

Les outils essentiels

Serveurs locaux

Un serveur local permet de simuler un environnement serveur sur votre machine. Ces serveurs incluent un interpréteur PHP, une base de données, et un serveur HTTP (comme Apache ou Nginx).

Outils recommandés :

- **XAMPP** :
  - Disponible sur Windows, macOS et Linux.
  - Inclut Apache, MySQL, PHP, et PhpMyAdmin.
  - Télécharger XAMPP.
- **WAMP (Windows) :**
  - Similaire à XAMPP, mais spécifique à Windows.
  - Inclut Apache, MySQL et PHP.
  - Télécharger WAMP.
- **MAMP (macOS et Windows) :**
  - Spécialement conçu pour macOS et Windows.
  - Inclut Apache, Nginx, MySQL et PHP.
  - Télécharger MAMP.
- **Laragon :**
  - Léger et rapide pour Windows.
  - Permet de créer des environnements de développement modulaires.
  - Télécharger Laragon.

Installation de XAMPP :

1. Téléchargez et installez XAMPP depuis le site officiel.
2. Lancez XAMPP et démarrez les modules Apache et MySQL.
3. Placez vos fichiers PHP dans le dossier **htdocs** (généralement situé dans **C:\xampp\htdocs**).
4. Accédez à vos fichiers via le navigateur en entrant **http://localhost/nom_du_fichier.php**.

Les outils essentiels

Éditeurs de texte et environnements de développement

Un bon éditeur de texte rend le développement plus fluide grâce à des fonctionnalités comme la coloration syntaxique, l'autocomplétion, et l'intégration avec des outils tiers.

Éditeurs de texte recommandés:

- **Visual Studio Code :**
  - Gratuit, léger et extensible avec des extensions.
  - Extensions utiles pour PHP :
  - PHP IntelliSense : Autocomplétion et documentation.
  - PHP Intelephense : Analyse avancée de code.
  - Xdebug : Pour le débogage.
  - Télécharger VS Code.
- **Sublime Text :**
  - Éditeur léger et rapide.
  - Support des extensions via Package Control.
  - Télécharger Sublime Text.
- **PhpStorm :**
  - IDE complet pour PHP.
  - Support avancé de PHP, bases de données et frameworks.
  - Payant, mais offre une version d'essai.
  - Télécharger PhpStorm.

Configurer Visual Studio Code pour PHP :

1. Installez PHP sur votre système :
2. Télécharger PHP.
3. Configurez la variable d'environnement pour que PHP soit accessible via le terminal.
4. Installez les extensions mentionnées ci-dessus dans VS Code.
5. Configurez un débogueur PHP avec Xdebug.

Les outils essentiels

Navigateur Web et outils de développement

Un navigateur web est essentiel pour tester vos applications. Les navigateurs modernes offrent des outils de développement pour déboguer, analyser et optimiser vos pages web.

Navigateurs recommandés :

- **Google Chrome :**
  - Rapide et fiable.
  - Inclut les DevTools (F12).
  - Extensions utiles :
    - ✓ Postman : Tester les API.
    - ✓ JSON Viewer : Visualiser les réponses JSON.
- **Mozilla Firefox :**
  - Open source et axé sur la confidentialité.
  - Les outils de développement sont puissants, surtout pour CSS et JavaScript.
- **Microsoft Edge :**
  - Basé sur Chromium, offre des DevTools similaires à Chrome.

Outils pour la gestion des bases de données

Pour manipuler des bases de données comme MySQL, il existe des outils graphiques qui simplifient la gestion.

Outils recommandés :

- **PhpMyAdmin :**
  - Inclus avec XAMPP/WAMP.
  - Permet de gérer des bases de données MySQL via une interface web.
  - URL par défaut : http://localhost/phpmyadmin.
- **Adminer :**
  - Plus léger que PhpMyAdmin.
  - Télécharger Adminer.
- **MySQL Workbench :**
  - Outil avancé pour concevoir et administrer des bases de données.
  - Télécharger MySQL Workbench.

Contrôle de version avec Git

Git est un système de contrôle de version qui permet de suivre les modifications de votre code.

Installation de Git :

Téléchargez et installez Git depuis git-scm.com.

Configurez Git :

```
git config --global user.name "VotreNom"
git config --global user.email "VotreEmail"
```

Utilisation Basique de Git :

Initialisez un dépôt :

```
git init
```

Ajoutez des fichiers au suivi :

```
git add fichier.php
```

Faites un commit :

```
git commit -m "Premier commit"
```

Connectez-vous à un dépôt distant (GitHub, GitLab, etc.) :

```
git remote add origin URL_DU_DEPOT
git push -u origin master
```

Les outils essentiels

Exemples pratiques

1. Installez XAMPP

2. Créez un fichier nommé **test.php** dans le dossier **htdocs** :

```php
<?php
echo "Serveur local opérationnel!";
?>
```

3. Ouvrez **http://localhost/test.php** dans un navigateur. Vous devriez voir le message "**Serveur local opérationnel!**".

Quiz

1. Quel outil est inclus avec XAMPP pour gérer les bases de données ?

    a) MySQL Workbench
    b) PhpMyAdmin
    c) HeidiSQL

2. Quelle commande Git est utilisée pour enregistrer des modifications localement ?

    a) git push
    b) git commit
    c) git clone

PHP

Bases de PHP

Qu’est-ce que PHP ?

PHP (Hypertext Preprocessor) est un langage de programmation côté serveur conçu pour créer des pages web dynamiques. Il est particulièrement apprécié pour sa simplicité, sa flexibilité, et sa compatibilité avec la majorité des systèmes d’exploitation et serveurs web.

Qu’est-ce que PHP ?

Historique de PHP

- **Création** : PHP a été développé en 1994 par Rasmus Lerdorf pour gérer son site personnel.
- **Évolution** : Initialement appelé "Personal Home Page Tools", PHP est devenu un langage complet avec l'introduction de PHP 3.0 en 1998.
- **Version actuelle** : La dernière version de PHP 8. x est PHP 8.3 , avec PHP 8.4 prévu pour une sortie en novembre 2024.

Définition de PHP

- PHP est un langage de script interprété qui s'exécute sur le serveur. Contrairement à JavaScript, qui s'exécute côté client, PHP génère du contenu (HTML, JSON, XML, etc.) envoyé au client (navigateur).

Qu’est-ce que PHP ?

Caractéristiques principales

Dynamisme :

PHP peut générer des pages web personnalisées selon les données saisies par l’utilisateur, les informations de la base de données ou d’autres facteurs.

Exemple :

```php
<?php
$nom = "Mahamane";
echo "Bonjour, $nom ! Bienvenue sur notre site.";
?>
```

Qu’est-ce que PHP ?

Caractéristiques principales

Intégration facile avec HTML :

PHP s’insère directement dans du code HTML, ce qui facilite la création de pages dynamiques.
Exemple :

```
<!DOCTYPE html>
<html>
<head>
    <title>Exemple PHP</title>
</head>
<body>
    <h1><?php echo "Bonjour, le monde !"; ?></h1>
</body>
</html>
```

Qu’est-ce que PHP ?

Caractéristiques principales

Compatibilité avec les bases de données:

PHP prend en charge une large gamme de bases de données, notamment :.

MySQL (le plus couramment utilisé).
PostgreSQL, SQLite, MongoDB, etc.

Qu'est-ce que PHP ?

Avantages de PHP

- **Open Source :**
  - Gratuit et soutenu par une grande communauté.
  - Facilité d'apprentissage :
  - Syntaxe simple, idéal pour les débutants.
- **Multiplateforme :**
  - Fonctionne sur Windows, Linux, macOS, etc.
- **Performances :**
  - Capable de gérer des projets à grande échelle comme Facebook (PHP modifié via HHVM).
- **Large Écosystème :**
  - Cadres (frameworks) comme Laravel, Symfony.
  - CMS comme WordPress, Joomla, Drupal.

Qu’est-ce que PHP ?

Les limites de PHP

- **Performance côté serveur :**
  - Moins rapide que des langages compilés comme C++.
  - Cependant, PHP 7 et PHP 8 ont grandement amélioré cet aspect.
- **Sécurité :**
  - Les mauvaises pratiques (par exemple, ne pas valider les entrées utilisateur) peuvent mener à des vulnérabilités comme l’injection SQL.

Fonctionnement de PHP

Lorsque vous accédez à une page **.php** via un navigateur :

- Le serveur web (Apache, Nginx) identifie le fichier comme un script PHP.
- L’interpréteur PHP traite le fichier.
- Le serveur envoie une réponse au navigateur sous forme de HTML.

Qu'est-ce que PHP ?

Les applications courantes de PHP

- **Sites dynamiques :**
  - Génération de pages personnalisées selon les utilisateurs.
- **Gestion des bases de données :**
  - Affichage et manipulation des données stockées dans MySQL.
- **API Web :**
  - PHP peut servir à créer des API RESTful qui renvoient des données JSON.
- **CMS (Systèmes de Gestion de Contenu) :**
  - PHP est le moteur derrière des systèmes comme WordPress.
- **Applications Web Complexes :**
  - E-commerce (Magento, WooCommerce).
  - Réseaux sociaux (Facebook à ses débuts).

Qu'est-ce que PHP ?

Exercice : Tester PHP

- **But : Créez un fichier PHP qui affiche une phrase dynamique basée sur l'heure actuelle.**

Code suggéré :

```php
<?php
$heure = date("H");
if ($heure < 12) {
    echo "Bonjour, il est matin !";
} elseif ($heure < 18) {
    echo "Bon après-midi !";
} else {
    echo "Bonsoir !";
}
?>
```

Instructions :

- Placez le fichier dans votre serveur local.
- Accédez-y via le navigateur et observez les résultats à différents moments de la journée.

Qu’est-ce que PHP ?

Quiz

- **Que signifie l'acronyme PHP ?**

  a) Personal Home Page
  b) Professional Hypertext Preprocessor
  c) Public HTML Parser

- **PHP est-il un langage côté serveur ou côté client ?**

  a) Serveur
  b) Client
  c) Les deux

Structure du code

PHP possède une structure de code simple et facile à apprendre, ce qui en fait un excellent langage pour les débutants et les développeurs expérimentés. Cette section explique comment organiser et structurer le code PHP, avec des exemples pratiques.

Structure du code

Les balises PHP

Le code PHP est encapsulé dans des balises spéciales :

```
<?php ... ?>
```

Exemple :

```
<?php
echo "Bonjour, le monde !";
?>
```

Autres styles de balises (moins courants) :

- Balises courtes : <? ... ?>
  Remarque : Elles doivent être activées dans le fichier de configuration PHP.
- Balises ASP : <% ... %>
  Obsolètes et non recommandées.

Structure du code

Organisation de base d'un script PHP

Un script PHP peut inclure :

1. Déclarations de variables
2. Structures conditionnelles
3. Boucles
4. Fonctions
5. Interactions avec une base de données

Structure du code

Les commentaires

Les commentaires sont essentiels pour rendre le code lisible et documenté.

Types de commentaires :

Commentaire sur une ligne :

```
// Ceci est un commentaire
```

Commentaire sur une ligne avec dièse :

```
# Ceci est aussi un commentaire
```

Commentaire multi-lignes :

```
/*
   Ceci est un commentaire
   sur plusieurs lignes
*/
```

Structure du code

Les commentaires

Exemple pratique :

```php
<?php
// Déclarer une variable
$nom = "Mahamane";

/*
   Afficher le contenu
   de la variable $nom
*/
echo "Bonjour, $nom !";
?>
```

Structure du code

Variables et constantes

1. Variables

- Les variables commencent par un signe **$** suivi d'un nom.
- Sensible à la casse ($Nom et $nom sont différents).
- Types dynamiques (pas besoin de déclarer leur type).

Exemple :

```php
<?php
$nom = "Ali";
$age = 25;

echo "Nom : $nom, Âge : $age";
?>
```

Structure du code

Variables et constantes

2. Constantes

- Les constantes ne changent jamais de valeur après leur définition.
- Déclarées avec **define()** ou **const**.

Exemple :

```php
<?php
define("SITE_WEB", "example.com");
echo "Bienvenue sur " . SITE_WEB;

// Avec const
const VILLE = "Niamey";
echo "Vous êtes à " . VILLE;
?>
```

Structure du code

Les structures de contrôle

1. Conditionnelles

Les structures conditionnelles permettent d'exécuter du code en fonction de certaines conditions.

Syntaxe **if** :

```php
<?php
$age = 18;

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

Structure du code

Les structures de contrôle

Syntaxe **switch** :

```php
<?php
$jour = "Lundi";

switch ($jour) {
    case "Lundi":
        echo "Début de semaine";
        break;
    case "Vendredi":
        echo "Fin de semaine";
        break;
    default:
        echo "Jour ordinaire";
}
?>
```

Structure du code

Les structures de contrôle

2. Boucles

Boucle **for** :

```php
<?php
for ($i = 1; $i <= 5; $i++) {
    echo "Nombre : $i <br>";
}
?>
```

Boucle **while** :

```php
<?php
$i = 1;
while ($i <= 5) {
    echo "Nombre : $i <br>";
    $i++;
}
?>
```

Structure du code

Les structures de contrôle

2. Boucles

Boucle **foreach** :

```php
<?php
$fruits = ["Pomme", "Banane", "Orange"];
foreach ($fruits as $fruit) {
    echo "Fruit : $fruit <br>";
}
?>
```

Fonctions et inclusion de fichiers

Les fonctions permettent de structurer le code en blocs réutilisables.

Fonctions et inclusion de fichiers

Définir et appeler une fonction

```php
<?php
function saluer($nom) {
    return "Bonjour, $nom !";
}

echo saluer("Mahamane");
?>
```

Fonctions avec paramètres par défaut

```php
<?php
function addition($a = 5, $b = 10) {
    return $a + $b;
}

echo addition();        // Affiche 15
echo addition(20, 30);  // Affiche 50
?>
```

Fonctions et inclusion de fichiers

Pour organiser le code, PHP permet d'inclure d'autres fichiers..

Fonctions et inclusion de fichiers

include

Génère un **avertissement** si le fichier n'est pas trouvé.

```php
<?php
include "header.php";
?>
```

Require

Génère une **erreur fatale** si le fichier n'est pas trouvé

```php
<?php
require "config.php";
?>
```

include_once / require_once

Garantit que le fichier n'est inclus qu'une seule fois.

```php
<?php
include_once "header.php";
?>
```

Fonctions et inclusion de fichiers

Organisation du code en PHP moderne

- Séparation du code
  - Code PHP dans des fichiers .php.
  - HTML dans des fichiers séparés ou intégrés via des templates.
- Respect des standards PSR (PHP Standards Recommendations)
  - PSR-1 : Standards de base pour le code PHP.
  - PSR-4 : Chargement automatique des classes.
- Utilisation des frameworks
  - Pour les projets complexes, utilisez des frameworks comme Laravel ou Symfony pour une meilleure structure.

Fonctions et inclusion de fichiers

Exemple pratique : Application simple

- But :
  - Créer une page affichant un message en fonction de l'heure.
- Code :
  - Fichier **functions.php** :

```php
<?php
function salutation() {
    $heure = date("H");
    if ($heure < 12) {
        return "Bonjour !";
    } elseif ($heure < 18) {
        return "Bon après-midi !";
    } else {
        return "Bonsoir !";
    }
}
?>
```

  - Fichier **index.php** :

```php
<?php
include "functions.php";
?>
<!DOCTYPE html>
<html>
<head>
    <title>Page PHP Structurée</title>
</head>
<body>
    <h1><?php echo salutation(); ?></h1>
</body>
</html>
```

Fonctions et inclusion de fichiers

Exercice

- Créez une fonction qui prend deux nombres et retourne leur produit. Testez-la.
- Utilisez une boucle pour afficher les 10 premières tables de multiplication.
- Divisez le code suivant en deux fichiers : un pour les fonctions et un autre pour l'affichage.

Fonctions et inclusion de fichiers

Quiz

- Quelle fonction est utilisée pour inclure un fichier tout en générant une erreur fatale s'il est absent ?
  a. include
  b. require
  c. require_once

- Quelle syntaxe est correcte pour déclarer une constante en PHP ?

  a. const NOM = "Valeur";
  b. $NOM = "Valeur";
  c. constant("NOM", "Valeur");

PHP

Manipulation des données

Formulaires HTML et PHP

Les formulaires sont essentiels pour interagir avec les utilisateurs. Ils permettent de collecter des données que PHP peut traiter. Cette section explique comment créer des formulaires HTML, récupérer les données en PHP, et sécuriser leur traitement.

Création d'un Formulaire HTML

Un formulaire HTML contient différents champs (texte, bouton, case à cocher, etc.) pour capturer les données de l'utilisateur. Ces données sont envoyées au serveur à l'aide des méthodes **GET** ou **POST**.

Exemple de formulaire simple

```
<!DOCTYPE html>
<html>
<head>
    <title>Formulaire Exemple</title>
</head>
<body>
    <form action="traitement.php" method="POST">
        <label for="nom">Nom :</label>
        <input type="text" id="nom" name="nom" required>
        <br><br>
        <label for="email">Email :</label>
        <input type="email" id="email" name="email" required>
        <br><br>
        <button type="submit">Envoyer</button>
    </form>
</body>
</html>
```

action : URL ou fichier où les données seront envoyées.
method : Méthode HTTP utilisée pour envoyer les données (POST ou GET).

Méthodes GET et POST

Méthode GET

- Envoie les données dans l'URL.
- Utilisée pour des recherches ou des données non sensibles.
- Limitation de taille (environ 2000 caractères).

Exemple :

Formulaire avec method="GET" :

```
<form action="traitement.php" method="GET">
    <label for="recherche">Recherche :</label>
    <input type="text" id="recherche" name="recherche">
    <button type="submit">Rechercher</button>
</form>
```

Résultat dans l'URL :
http://localhost/traitement.php?recherche=motclé

Méthodes GET et POST

Méthode POST

- Envoie les données dans le corps de la requête HTTP.
- Plus sécurisé que GET (les données ne sont pas visibles dans l'URL).
- Pas de limite de taille pour les données.

Exemple :

Formulaire avec method= »POST" :

```
<form action="traitement.php" method="POST">
    <label for="nom">Nom :</label>
    <input type="text" id="nom" name="nom">
    <button type="submit">Envoyer</button>
</form>
```

Récupérer les données en PHP

PHP fournit des superglobales pour accéder aux données des formulaires :

- **$_GET** : Données envoyées par la méthode GET.
- **$_POST** : Données envoyées par la méthode POST.
- **$_REQUEST** : Contient à la fois les données GET et POST.

Exemple :
Fichier **traitement.php** :

```php
<?php
if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    $nom = $_POST['nom'];
    $email = $_POST['email'];

    echo "Nom : " . htmlspecialchars($nom) . "<br>";
    echo "Email : " . htmlspecialchars($email);
}
?>
```

$_SERVER['REQUEST_METHOD'] : Vérifie si le formulaire a été soumis avec POST.
htmlspecialchars() : Protège contre les injections HTML.

Validation et sécurisation des formulaires

1. Validation côté serveur

Assurez-vous que les données reçues sont valides avant de les traiter.

Exemple :

```php
<?php
if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    $nom = trim($_POST['nom']);
    $email = trim($_POST['email']);

    if (empty($nom)) {
        echo "Le champ Nom est requis.";
    } elseif (!filter_var($email, FILTER_VALIDATE_EMAIL)) {
        echo "L'email n'est pas valide.";
    } else {
        echo "Données valides. Merci, $nom !";
    }
}
?>
```

Validation et sécurisation des formulaires

2. Sécurisation

Nettoyage des données :

```php
$nom = htmlspecialchars(strip_tags($nom));
```

Échapper les caractères spéciaux avant d'insérer dans une base de données :

```php
$nom = mysqli_real_escape_string($connexion, $nom);
```

Utilisation avancée : envoi et traitement

Formulaire avec plusieurs champs

```
<form action="traitement.php" method="POST">
    <label for="nom">Nom :</label>
    <input type="text" id="nom" name="nom" required>
    <br><br>
    <label for="email">Email :</label>
    <input type="email" id="email" name="email" required>
    <br><br>
    <label for="message">Message :</label>
    <textarea id="message" name="message"></textarea>
    <br><br>
    <label>
        <input type="checkbox" name="newsletter" value="oui"> S'inscrire à la news
    </label>
    <br><br>
    <button type="submit">Envoyer</button>
</form>
```

Formulaires HTML et PHP

Utilisation avancée : envoi et traitement

Traitement des données

```php
<?php
if ($_SERVER['REQUEST_METHOD'] == 'POST') {
    $nom = htmlspecialchars($_POST['nom']);
    $email = htmlspecialchars($_POST['email']);
    $message = htmlspecialchars($_POST['message']);
    $newsletter = isset($_POST['newsletter']) ? 'Oui' : 'Non';

    echo "Nom : $nom<br>";
    echo "Email : $email<br>";
    echo "Message : $message<br>";
    echo "Inscription à la newsletter : $newsletter";
}
?>
```

Téléchargement de fichiers

PHP permet aussi de gérer les fichiers envoyés via un formulaire.

Formulaire pour envoyer un fichier

```
<form action="upload.php" method="POST" enctype="multipart/form-data">
    <label for="fichier">Choisir un fichier :</label>
    <input type="file" id="fichier" name="fichier">
    <button type="submit">Télécharger</button>
</form>
```

Traitement du fichier

```
<?php
if ($_SERVER['REQUEST_METHOD'] == 'POST' && isset($_FILES['fichier'])) {
    $nomFichier = $_FILES['fichier']['name'];
    $cheminTemp = $_FILES['fichier']['tmp_name'];
    $dossierDestination = "uploads/";

    if (move_uploaded_file($cheminTemp, $dossierDestination . $nomFichier)) {
        echo "Fichier téléchargé avec succès : $nomFichier";
    } else {
        echo "Échec du téléchargement.";
    }
}
?>
```

Exercice pratique

Formulaire de Contact

- Créez un formulaire avec les champs suivants :
  - Nom
  - Email
  - Sujet
  - Message
- Traitez les données en PHP :
  - Validez les champs (tous obligatoires).
  - Affichez un message de confirmation.

Quiz

- Quelle méthode d'envoi est utilisée pour inclure les données dans l'URL ?
  - GET
  - POST
  - PUT
- Quelle fonction PHP est utilisée pour protéger contre les injections HTML ?
  - trim()
  - htmlspecialchars()
  - mysqli_escape_string()

Gestion des variables globales

- ….

Gestion des fichiers

- ….

PHP

Bases de données avec PHP

Introduction à MySQL

- ….

Connexion PHP/MySQL

- ….

CRUD en PHP

- ….

PHP

Gestion des sessions et cookies

Sessions en PHP

- ….

Cookies

- ….

PHP

PHP avancé

Programmation Orientée Objet (POO)

- ….

Traitement des API

- ….

Sécurité en PHP

- ….

PHP

Travail avec les frameworks PHP

Pourquoi utiliser un Framework ?

- ….

Introduction à Laravel

- ….

Autres frameworks

- ….

PHP

Projet final

Créer un projet complet, par exemple :

- Système de gestion des utilisateurs : Inscription, connexion, modification de profil.
- Blog : Ajout, édition, et suppression d'articles.
- Boutique en ligne : Gestion des produits, panier, commandes.