

# PROGRAMMATION PHP-8 – MYSQL-8

## PRESENTATION GENERALE

### L2S – 7-9 AVRIL 2025

### 3 JOURNEES

Prof : Bertrand Liaudet  
Supports de cours en ligne : [http://bliaudet.free.fr/rubrique.php3?id\\_rubrique=276](http://bliaudet.free.fr/rubrique.php3?id_rubrique=276)

## SOMMAIRE

<b>Sommaire .....</b>	<b>1</b>
<b>PRESENTATION GENERALE.....</b>	<b>3</b>
<b>Plan des 3 journées .....</b>	<b>3</b>
J-1 : PHP sans MySQL.....	3
J-2 : PHP MySQL en MVC de base – Travail sur une étude de cas .....	3
J-3 : Suite de J-2 .....	3
<b>Évaluation.....</b>	<b>4</b>
<b>Le Web dynamique : concepts .....</b>	<b>5</b>
Technique pour dynamiser la page HTML.....	5
Web 1.0 : Web statique (côté client, front end, front): que la page HTML-CSS .....	5
Web 1.0 : Web dynamique côté client (front end, front) : HTML-CSS + JavaScript .....	5
Web 2.0 : Web dynamique côté client (front end, front) : HTML-CSS-JavaScript + AJAX.....	6
Web 2.0 « standard » : Web dynamique côté client (front end, front) : HTML-CSS-JavaScript-AJAX + langage serveur.....	6
<b>Architecture d’une application de base de données pour le web .....</b>	<b>7</b>
Architecture client-serveur.....	7
Schéma général.....	7
Exemple de chemin web standard .....	8
Chemin WEB standard : .....	9
Chemin WEB via AJAX : .....	10
Chargement d’un fichier HTML sur la machine : .....	11
Les langages de programmation et programmes utilisés : .....	12
Langages côté client.....	12
Applications côté client.....	12
Langages côté serveur .....	12
Applications côté serveur .....	12
CRUD.....	13
Un usage majeur du PHP : PHP-MySQL : le PHP et les Bases de données. ....	13
CRUD : CREATE, READ, UPDATE, DELETE .....	13
Une application WEB tourne autour du CRUD .....	14
MVC .....	15
MVC veut dire Modèle – Vue – Controleur .....	15
Ça revient à l’architecture standard de tout programme propre et « pro » : .....	15
<b>Le langage de Script PHP .....</b>	<b>16</b>
Historique : 30 ans ! le vieux code PHP est obsolète ! .....	16
PHP.....	16
MySQL.....	16
Principe du PHP : produire du code HTML à afficher .....	17
Principes généraux.....	17
HTML “pur” .....	17

Balise PHP .....	17
<b>Environnement de travail.....</b>	<b>18</b>
Visual Studio Code : .....	18
WAMP : Principes : serveur WEB, serveur de BD, phpMyAdmin.....	18

Edition Avril 2025

# PRESENTATION GENERALE

## Plan des 3 journées

Principe général :

½ journée théorique

½ journée pratique

Si la théorie avance bien, on peut avancer la pratique le matin.

### J-1 : PHP sans MySQL

- 1) Introduction
- 2) Bases du langage

### J-2 : PHP MySQL en MVC de base – Travail sur une étude de cas

- 3) \$\_GET – Faille XSS : 2 exercices.
- 4) L'application Posts-Users-Comments en HTML
- 5) L'application Posts-Users-Comments en PHP
- 6) La BD
  - Créer la BD et les tables
  - Peupler la BD
  - SELECT des Users
- 7) CRUD : R : SELECT User
  - Connexion à la BD en PHP
  - SELECT dans la BD en PHP
  - L'application Posts-Users-Comments en PHP avec un SELECT

### J-3 : Suite de J-2

- 8) CRUD : D : DELETE Users, Comments, Delete Posts – Faille Injection SQL
- 9) \$\_POST et Formulaire – Faille XSS
- 10) CRUD : C : INSERT Users
- 11) CRUD : U : UPDATE Users

## Évaluation

- Contrôle continu : rendu par mail des travaux de la journée : en fin de chaque journée.
- Notation de l'avancement des exercices en cours de séance.
- Partiel terminal (date non déterminée, après le cours) sous forme de QCM de 20 questions.

⇒ Le but est que vous ayez compris les bases du PHP moderne et de la programmation WEB en général.

## Le Web dynamique : concepts

### Technique pour dynamiser la page HTML

#### Web 1.0 : Web statique (côté client, front end, front): que la page HTML-CSS

- ⇒ Le **HTML-CSS** permet de créer des pages WEB.
- ⇒ **Seule, c'est statique** : pas d'interaction avec les utilisateurs
- ⇒ **Seul, ça n'existe plus : c'est pour jouer ou pédagogique.**

#### Web 1.0 : Web dynamique côté client (front end, front) : HTML-CSS + JavaScript

- ⇒ Le **HTML-CSS-JavaScript** permet de dynamiser la page HTML
- ⇒ On peut interagir avec un utilisateur **SANS** interagir avec un serveur.
- ⇒ **Seule : ça n'existe plus : c'est pour jouer ou pédagogique.**

## Web 2.0 : Web dynamique côté client (front end, front) : HTML-CSS-JavaScript + AJAX

- ⇒ Le HTML-CSS-JavaScript + Ajax permet de dynamiser la page HTML
- ⇒ On peut interagir avec un utilisateur ET interagir avec un serveur avec AJAX.
- ⇒ AJAX sous-entend une Architecture Client-Serveur : on va y venir.
- ⇒ **Angular et React permettent de faire ça.** AJAX permet l'accès à une BD, en général via une API.
- ⇒ **Ca permet de réaliser des application WEB « single page ».**

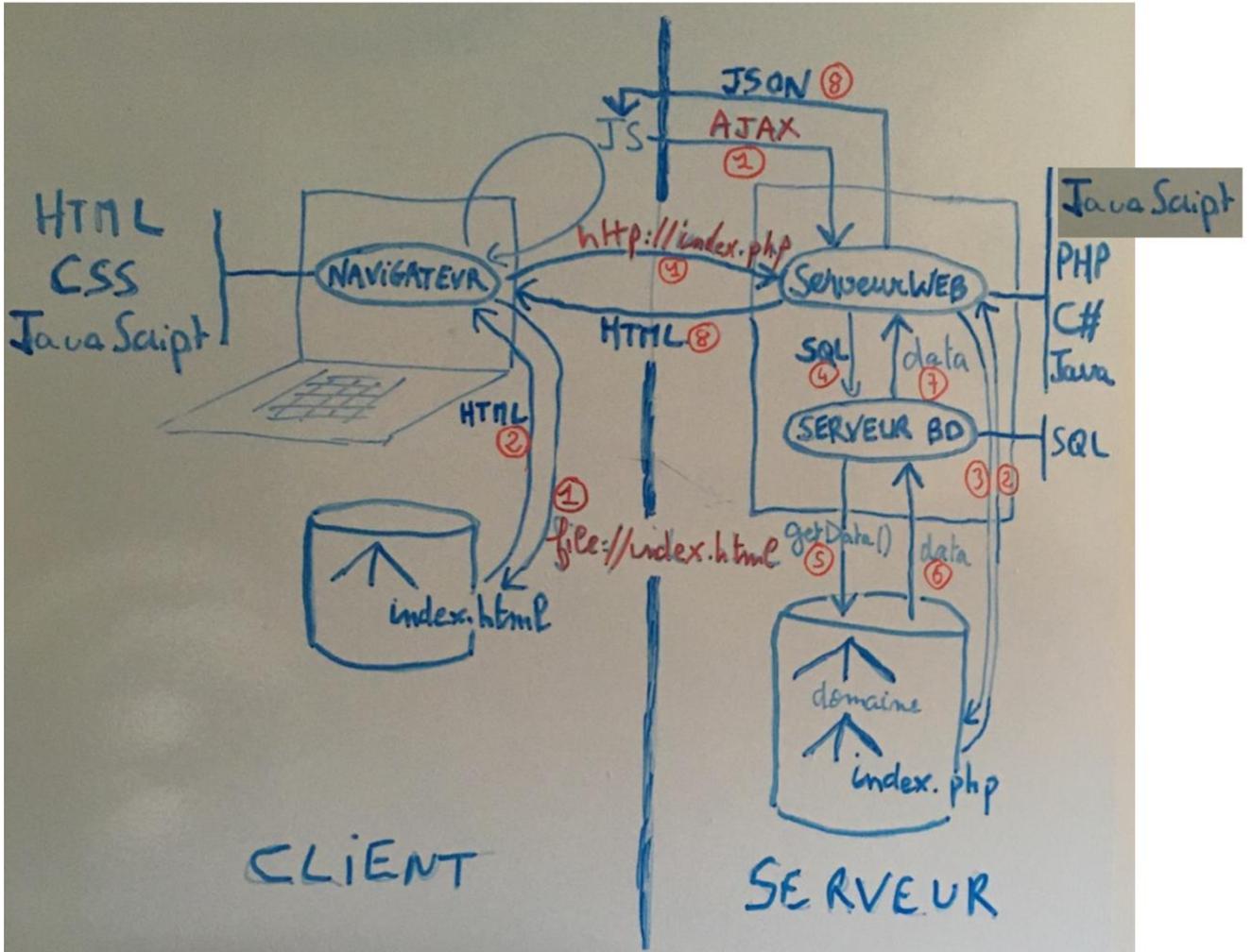
## Web 2.0 « standard » : Web dynamique côté client (front end, front) : HTML-CSS-JavaScript-AJAX + langage serveur

- ⇒ Le HTML-CSS-JavaScript-Ajax + langage serveur : permet de réaliser une **application WEB standard adaptable à un téléphone.**
- ⇒ On peut interagir avec un utilisateur ET interagir avec un serveur.
- ⇒ Ça sous-entend une Architecture Client-Serveur : on va y venir.
- ⇒ **Le PHP permet de faire ça (comme le Java, le C#, le Python, Node, etc.)**

# Architecture d'une application de base de données pour le web

## Architecture client-serveur

### Schéma général

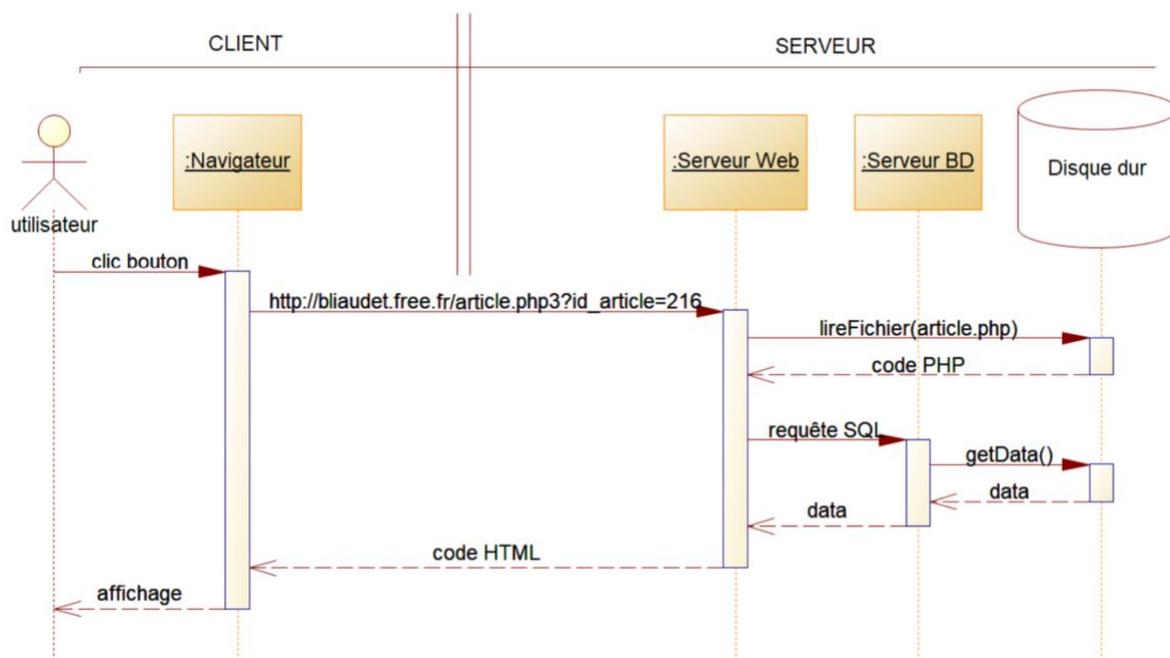


## Exemple de chemin web standard

### Exemple :

[http://bliaudet.free.fr/article.php3?id\\_article=485](http://bliaudet.free.fr/article.php3?id_article=485)

- Je vais chercher le fichier article.php3 qui se trouve sur le domaine bliaudet.free.fr
- Je passe l'id\_article 485 en paramètre
- Le serveur web récupère le fichier sur le disque dur
- Le serveur web demande l'article 485 à la BD
- Le serveur web fabrique la page HTML et la renvoie au client



## Chemin WEB standard :

### ➤ 1 :

- Le navigateur fournit une URL : on demande un fichier : <http://bliaudet.free.fr>
- En fait c'est <http://bliaudet.free.fr/index.php3> (c'est vieux !)
- Le fichier c'est index.php3
- Sur la machine serveur identifiée par <http://bliaudet.free.fr> : il y a un programme qui tourne : c'est un serveur WEB. On communique avec ce serveur et on lui dit de traiter le fichier dont on a donné le nom.

### ➤ 2-3 :

- Le serveur WEB (c'est un programme) récupère le fichier sur son disque dur.
- Le serveur WEB lit les instructions dans le fichier et les exécute. Le but est à la fin de produire du code HTML

### ➤ 4 :

- Une instruction est une requête CRUD (INSERT, ou SELECT ou UPDATE ou DELETE).
- Cette requête est envoyée au serveur de BD

### ➤ 5 :

- Le serveur de BD exécute cette requête et récupère les données (SELECT) ou modifie les données dans la BD (INSERT, UPDATE, DELETE).

### ➤ 6 :

- Si c'est un SELECT (par exemple), le serveur de BD récupère les données sur le disque dur.

### ➤ 7 :

- Le serveur de BD renvoie les données au serveur WEB
- Le serveur WEB continue de lire les instructions du fichier.

### ➤ 8 :

- Quand il a fini d'exécuter les instructions du fichier, le serveur WEB renvoie au client le code HTML qu'il a construit.

## **Chemin WEB via AJAX :**

➤ **0 :**

- Le navigateur fait tourner du code JavaScript

➤ **1 :**

- Le code JavaScript envoie une requête AJAX : c'est équivalent au 1 précédent

➤ **2 à 7 :**

- La même chose que précédemment

➤ **8 :**

Quand il a fini d'exécuter les instructions du fichier, le serveur WEB renvoie, en général, des données au format JSON

Le JavaScript du navigateur s'occupe d'actualiser la page du navigateur avec les données reçues.

## Chargement d'un fichier HTML sur la machine :

### ➤ 1 :

- On peut donner comme URL un nom de fichier sur sa machine :  
[file:///Users/vous/Desktop/votre\\_fichier.html](file:///Users/vous/Desktop/votre_fichier.html)

### ➤ 2 :

- Le navigateur récupère le fichier
- Le navigateur traite le code du fichier et l'affiche.
  - ⇒ Ca ne sert à tester du code HTML.
  - ⇒ Ca sert aussi à lire des pdf, des images, des audios, des vidéos.

## Les langages de programmation et programmes utilisés :

### Langages côté client

- HTML, CSS, JavaScript

### Applications côté client

- **Navigateurs** : Firefox, Internet Explorer, Safari, Opera, etc.

### Langages côté serveur

- SQL et PHP (ou C#, Java, Python, Ruby, etc.).

### Applications côté serveur

- **Un serveur HTTP** (ou serveur web) : Apache (PHP), Tomcat (Java, JEE), IIS (C#, .Net)
- **Un serveur de BD** : MySQL (ou Oracle, SQL-Server, PostgreSQL, MariaDB, SQLite, etc.)  
⇒ **Remarque** : un serveur c'est une machine qui fait fonctionner un programme qu'on appelle aussi un serveur (la machine est un serveur parce qu'elle contient un programme serveur). Un programme serveur est un programme qui tourne 24H sur 24 et qui a comme caractéristique de ne pas avoir d'interface utilisateur. Seuls d'autres programmes peuvent communiquer avec lui. Un programme serveur s'appelle **serveur** ou **daemon**.  
⇒ Le serveur mysql sur PC s'appelle mysqld, « d » pour daemon.

## CRUD

### Un usage majeur du PHP : PHP-MySQL : le PHP et les Bases de données.

- MySQL : une application WEB travaille avec les données de la BD.
- La BD c'est : des tables (comme une table excel).

Exemple :

id_etudiant	nom_etudiant	email_etudiant	date_de_naissance_etudiant
1	Toto	toto@mail.com	1/1/2001
2	Tata	<a href="mailto:tata@mail.com">tata@mail.com</a>	10/6/2002
3	Titi	titi@mail.com	5/12/2003

### CRUD : CREATE, READ, UPDATE, DELETE

- Ça correspond à 4 choses qu'on fait avec une table :
  - ⇒ **CREATE** : c'est en réalité un **INSERT** qui permet d'ajouter une ligne dans la table.
  - ⇒ **READ** : c'est en réalité un **SELECT** qui permet de sélectionner des lignes dans la table.
  - ⇒ **UPDATE** : c'est bien un UPDATE qui permet de modifier des lignes dans la table.
  - ⇒ **DELETE** : c'est bien un DELETE qui permet de supprimer des lignes dans la table.

## Une application WEB tourne autour du CRUD

- Ce qu'on fait sur une application WEB c'est souvent une série de CRUD.
- Ce n'est pas théorique ! Les framework modernes, comme Laravel, tournent autour du CRUD.

➤ *Exemple :*

**Créer son compte** : c'est un **C pour INSERT** d'un nouvel utilisateur dans la table des USERS.

**Modifier son mot de passe** : c'est un **U pour UPDATE** de la ligne de nous dans la table des USERS.

**Afficher des produits** : c'est un **R pour SELECT** des lignes de la table PRODUITS.  
Etc.

### MVC veut dire Modèle – Vue – Contrôleur

- C'est l'architecture standard d'une application WEB et d'un programme en général.
- Ce n'est pas théorique ! Les framework modernes, comme Laravel, tournent autour du CRUD.

⇒ **Le modèle** : ce sont les informations fournies par l'utilisateur et les outils qui vont nous permettre de manipuler les tables dont on aura besoin.

⇒ **La vue** : c'est l'affichage final qu'on produit : ici la page HTML.

⇒ **Le contrôleur** : c'est le programme qui charge le modèle, fait les calculs qu'il souhaite et qui pour finir appelle la vue qui se charge de l'affichage.

### Ca revient à l'architecture standard de tout programme propre et « pro » :

- 1) **Lire les données** du programmes (**le modèle**).
- 2) **Traiter les données** du programme et produire les données résultats : **le contrôleur**.
- 3) **Afficher les résultats**.

### Historique : 30 ans ! le vieux code PHP est obsolète !

#### PHP

- 1995 – Création du PHP public.
  - **1998-2004 – PHP 3, 4 et MySQL 4.x : L'essor du web dynamique**
  - 2004-2018 – 5 : complètement obsolète et dépassé.
  - 2015-2022 – PHP 7 a marqué une énorme évolution par rapport à PHP 5, mais il est aujourd'hui considéré comme obsolète et non moderne.
  - **2020 – PHP 8 : le PHP moderne depuis 2020**
- Aujourd'hui, le PHP est utilisé dans des **frameworks** qui le rendent « pro » :
- ⇒ Macro-framework : **Laravel**
  - ⇒ Micro-framework : **Slim**

**Écrire du PHP « à la papa » en 2025 n'a aucun sens !**

#### MySQL

- 1995 – Création de MySQL
- 2000 – MySQL devient open source sous licence GPL.
- 2005 – MySQL-5 - Sun Microsystems rachète MySQL pour 1 milliard \$.
- 2010 – Oracle rachète Sun pour 7,5 milliards \$ et prend le contrôle de MySQL.
- 2013 – Création de **MariaDB** par les anciens de MySQL
- **2018 – Sortie de MySQL 8.0, la version actuelle et moderne.**

## Principe du PHP : produire du code HTML à afficher

### Principes généraux

- Le PHP consiste à produire du code HTML à afficher.
- Les fichiers PHP sont des scripts d'extension : « .php »
- Le script est interprété par un serveur, le serveur WEB, qui lit le script et produit le code HTML à renvoyer au client qui à fait la demande.

### HTML “pur”

Un fichier PHP peut ne contenir que du HTML : dans ce cas, la traduction consiste à recopier. Le fichier .php est donc constitué de balises HTML.

### Balise PHP

- Un fichier PHP peut et va réalité, toujours contenir aussi des instructions de programmation classique (des tests, des boucles, des fonctions, des variables, des tableaux, des accès à la BD, etc.).
- Les instructions PHP seront placées dans une balise PHP : <php ? ?>

Exemple :

```
<?php
$nom = "TOTO" ; # $nom est une variable
$nom = ucfirst(strtolower($nom)); // Résultat : "Toto"
?>
<h2>Résultats : </h2>
<p>nom: <?php echo $nom; /* on affiche le nom */ ?> </p>
```

## Environnement de travail

### Visual Studio Code :

- Pour écrire notre code, on utilise Visual Studio Code :
- Il faut charger des packages pour faciliter l'écriture du code. Au minimum :
  - ⇒ French Language Pack For Visual Studio Code
  - ⇒ PHP Intellephense
  - ⇒ Prettier Code Formater
  - ⇒ Prettier SQL VSCode
- Avec ça, les bugs de syntaxe apparaissent en coloration syntaxique.
- On peut formater le code avec maj-alt-f
- VS Code s'utilise avec des dossiers : pas directement avec un fichier.

### WAMP : Principes : serveur WEB, serveur de BD, phpMyAdmin

- Pour coder du PHP, on a besoin d'un serveur WEB et d'un serveur de BD.
- Une solution simple et courante est d'installer un environnement qui installe les 2. Il y en a plein :
  - ⇒ WAMP (MAMP sur Mac)
  - ⇒ XAMPP
  - ⇒ Laravel
  - ⇒ Etc.
- Il est pratique que ces environnements permettent un accès à **phpMyAdmin** qui est une application WEB permettant de gérer la BD.
- Pour ce cours, on utilisera WAMP.