

	<h1>Concevoir un site web avec PHP et MySQL</h1>	Réalisé le :	28-12-2024
		Modifié le :	28/12/2024

Nombre de pages :  
61

# Concevoir un site web avec PHP et MySQL

<b>1. PARTIE 1 - FAIRE LES PREMIERS PAS EN PHP</b>	<b>4</b>
1.1. TIRER UN MAXIMUM DE CE COURS	4
1.1.1. Rencontrez vos professeurs ✓	4
1.1.2. Mettez-vous à niveau en HTML et CSS si besoin ✓	4
1.1.3. Pratiquer en suivant le projet fil rouge	4
1.2. DECOUVRIR LE FONCTIONNEMENT D'UN SITE ECRIT EN PHP	4
1.2.1. Faire la différence entre site statique et dynamique	5
1.2.1.1. Découvrez le principe d'un site statique	5
1.2.1.2. Découvrez le principe d'un site dynamique	5
1.2.2. Comprenez le fonctionnement d'un site web	5
1.2.2.1. Consultez un site statique	6
1.2.2.2. Consultez un site dynamique	6
1.2.3. Exploitez les langages du Web	7
1.2.3.1. Utilisez HTML et CSS pour un site statique	7
1.2.3.1.1. HTML	7
1.2.3.1.2. CSS	7
1.2.3.2. Ajoutez PHP et MySQL pour un site dynamique	8
1.2.3.2.1. PHP	8
1.2.3.2.2. MySQL	8
1.2.3.2.3. SQL	8
1.2.4. En résumé	10
1.2.5. Timeline des principales technologies du web	10
1.3. PREPARER L'ENVIRONNEMENT DE TRAVAIL	10
1.3.1. Ayez en tête les outils de base pour créer un site statique	11
1.3.1.1. Un éditeur de texte	11
1.3.1.2. Un navigateur web	11
1.3.2. Utilisez les outils pour créer un site dynamique	11
1.3.2.1. Apache	11
1.3.2.2. PHP	11
1.3.2.3. MySQL	11
1.3.3. Installez XAMPP sous Windows	12
1.3.3.1. Installation de Xampp	12
1.3.3.2. Test de l'installation de Xampp	14
1.3.4. Installez XAMPP sous macOS	15
1.3.4.1. Ne concerne que les utilisateurs sous Mac ✗	15
1.3.5. Installez XAMPP sous Linux ★	15
1.3.5.1. Installation de Xampp	15
1.3.5.2. Démarrage/ Arrêt de Xampp	17
1.3.5.3. Test de Xampp	17
1.3.6. Découvrez le serveur PHP intégré	17
1.3.6.1. Code PHP	17
1.3.7. Utiliser un bon éditeur de texte	18
1.3.7.1. Code HTML	18
1.3.7.2. Visual Studio Code	19
1.3.7.3. PHPStorm	19
1.3.8. En résumé	21
1.4. ECRIRE LE PREMIER SCRIPT	21

	<h1>Concevoir un site web avec PHP et MySQL</h1>	Réalisé le :	28-12-2024
Nombre de pages : 61		Modifié le :	28/12/2024

1.4.1. Utilisez des balises PHP	21
1.4.1.1. Reconnaître la forme d'une balise PHP	22
1.4.1.1.1. Il existe d'autres balises pour utiliser du PHP	22
1.4.1.2. Insérer une balise PHP au milieu du code HTML	23
1.4.1.2.1. On peut placer une balise PHP n'importe où dans le code	23
1.4.2. Affichez du texte	24
1.4.2.1. Utilisez l'instruction echo	24
1.4.2.2. Enregistrez une page PHP	26
1.4.2.2.1. Fichier affichertexte.php	26
1.4.2.3. Testez une page PHP	27
1.4.3. Commentez votre code	29
1.4.3.1. Faites des commentaires monoligne	30
1.4.3.1.1. Commentaire monoligne style Unix	30
1.4.3.2. Faites des commentaires multilignes	30
1.4.4. En résumé	31
1.5. CONFIGURER PHP POUR VISUALISER LES ERREURS	31
1.5.1. Configurez PHP pour afficher les erreurs	31
1.5.1.1. Localisez le fichier de configuration PHP du serveur web	31
1.5.1.2. Modifiez le fichier de configuration PHP	34
1.5.2. Testez l'affichage des erreurs	35
1.5.3. En résumé	36
1.6. QUIZ : PREMIERS PAS EN PHP	37
<b>2. PARTIE 2 - REALISEZ UN SITE WEB DYNAMIQUE AVEC PHP</b>	<b>39</b>
2.1. DECRIRE LES ELEMENTS DU PROJET A L'AIDE DE VARIABLES	39
2.1.1. Comprendre ce qu'est une variable	40
2.1.1.1. Donner toujours un nom et une valeur aux variables	40
2.1.1.2. Découvrir les différents types de variables	41
2.1.1.2.1. PHP est un langage de programmation à typage dynamique faible	41
2.1.2. Affecter une valeur à une variable	42
2.1.3. Utiliser les types de données	43
2.1.3.1. Le type string (chaîne de caractères)	43
2.1.3.2. Le type int (nombre entier)	44
2.1.3.3. Le type float (nombre décimal)	44
2.1.3.4. Le type bool (booléen)	45
2.1.3.5. Une variable vide avec NULL	45
2.1.4. Afficher le contenu d'une variable	45
2.1.5. Concaténer une variable	45
2.1.5.1. Effectuez l'interpolation avec des guillemets doubles	45
2.1.5.2. Concaténez avec des guillemets simples	46
2.1.5.3. To do : les opérateurs de chaînes de caractères	47
2.1.6. Faire des calculs simples	47
2.1.6.1. Les opérations de base : addition, soustraction...	47
2.1.6.2. Le modulo	47
2.1.7. En résumé	48
2.2. ADAPTER LE COMPORTEMENT DE L'APPLICATION A L'AIDE DES CONDITIONS	48
2.2.1. Appropriiez-vous la structure de base : if... else	49
2.2.1.1. Retenir les symboles à connaître	49
2.2.1.1.1. To do : les opérateurs PHP en carte heuristique (Xmind)	49
2.2.1.1.2. To do : Précédence des opérateurs en PHP	49
2.2.1.2. Utiliser la structure if... else	49

	<h1>Concevoir un site web avec PHP et MySQL</h1>	Réalisé le :	28-12-2024
Nombre de pages : 61		Modifié le :	28/12/2024

2.2.1.3. Étudier le cas des booléens	52
2.2.1.4. Poser des conditions multiples	53
2.2.1.4.1. Premier exemple avec &&	53
2.2.1.4.2. Deuxième exemple avec	53
2.2.1.5. Utiliser cette astuce bonus	54
2.2.2. Utilisez la condition switch pour optimiser votre code	54
2.2.3. Découvrez les ternaires : des conditions condensées	57
2.2.4. En résumé	58
2.3. AFFICHER UNE LISTE DE RECETTES A L'AIDE DES BOUCLES	59
2.4. ORGANISER LES DONNEES A L'AIDE DES TABLEAUX	59
2.5. EXPLOITER TOUTE LA PUISSANCE DES FONCTIONS PHP !	59
2.6. AU SECOURS ! MON SCRIPT PLANTE !	59
2.7. ORGANISER LES PAGES DU SITE EN BLOCS FONCTIONNELS	59
2.8. QUIZ : REALISER UN SITE WEB DYNAMIQUE AVEC PHP	59
<b>3. PARTIE 3 - TRANSMETTRE DES DONNEES DE PAGE EN PAGE</b>	<b>59</b>
3.1. ÉCOUTER LA REQUETE DES UTILISATEURS GRACE AUX URL	59
3.2. ADMINISTRER DES FORMULAIRES DE FAÇON SECURISEE	59
3.3. ACTIVER LE PARTAGE DE FICHIERS	59
3.4. IMPLEMENTER UN SYSTEME DE CONNEXION	59
3.5. CONSERVER DES DONNEES GRACE AUX SESSIONS ET AUX COOKIES	59
3.6. QUIZ : TRANSMETTRE DES DONNEES DE PAGE EN PAGE	59
<b>4. PARTIE 4 - STOCKEZ DES INFORMATIONS DANS UNE BASE DE DONNEES</b>	<b>59</b>
4.1. TRAVAILLER AVEC UNE BASE DE DONNEES	60
4.2. METTRE EN PLACE UNE BASE DE DONNEES AVEC PHPMYADMIN	60
4.3. ACCEDER AUX DONNEES EN PHP AVEC PDO	60
4.4. AJOUTER, MODIFIER ET SUPPRIMER DES RECETTES !	60
4.5. AJOUTER DES COMMENTAIRES GRACE AUX JOINTURES SQL	60
4.6. ALLER PLUS LOIN	60
4.7. QUIZ : UTILISER UNE BASE DE DONNEES	60
4.8. LE CERTIFICAT DE REUSSITE	60
<b>5. ANNEXES</b>	<b>60</b>
5.1. TABLE ASCII SUR 7 BITS	60
<b>6. ALLER PLUS LOIN</b>	<b>60</b>
6.1. COURS	60
6.2. CERTIFICATIONS TOSA	61
<b>7. END</b>	<b>61</b>

	<h1>Concevoir un site web avec PHP et MySQL</h1>	Réalisé le :	28-12-2024
		Modifié le :	28/12/2024

Nombre de pages :  
61

## 1. Partie 1 - Faire les premiers pas en PHP

### 1.1. Tirer un maximum de ce cours

#### 1.1.1. Rencontrez vos professeurs

#### 1.1.2. Mettez-vous à niveau en HTML et CSS si besoin

#### 1.1.3. Pratiquer en suivant le projet fil rouge

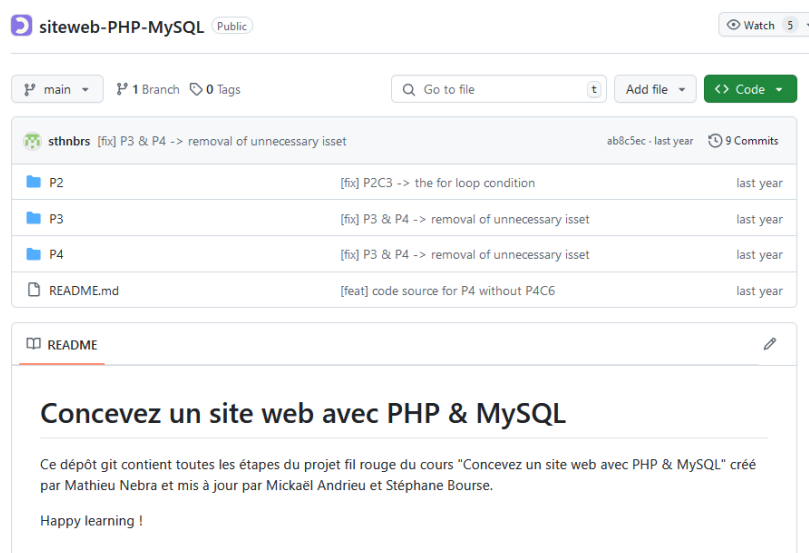
L'objectif de ce cours est de vous permettre de réaliser des sites web dynamiques, pas à pas.

Pour cela, nous avons mis au point dans ce cours un projet fil rouge. Cela signifie que vous allez avancer chapitre par chapitre en apprenant comment **réaliser un site web dynamique de partage de recettes de cuisine**. Chaque chapitre vous donnera des clés supplémentaires pour avancer dans ce projet pratique



Vous pouvez trouver l'ensemble de fichiers liés au projet fil rouge dans notre [repo Github](https://github.com/OpenClassrooms-Student-Center/siteweb-PHP-MySQL)<sup>1</sup>.

<https://github.com/OpenClassrooms-Student-Center/siteweb-PHP-MySQL.git>



### 1.2. Découvrir le fonctionnement d'un site écrit en PHP



Qu'est-ce qu'un serveur et un client ? Comment rend-on son site dynamique ? Et que signifient PHP et MySQL ?

<sup>1</sup> <https://github.com/OpenClassrooms-Student-Center/siteweb-PHP-MySQL>

	<h1>Concevoir un site web avec PHP et MySQL</h1>	Réalisé le :	28-12-2024
Nombre de pages : 61		Modifié le :	28/12/2024

## 1.2.1. Faire la différence entre site statique et dynamique

On considère qu'il existe deux types de sites web :

1. Les sites **statiques**.
2. Et les sites **dynamiques**.

### 1.2.1.1. Découvrez le principe d'un site statique

Un site statique est réalisé uniquement à l'aide des langages HTML et CSS.

Il fonctionne très bien, **mais son contenu ne peut pas être mis à jour automatiquement : il faut que le webmaster modifie le code source pour y ajouter des nouveautés.**

Ce n'est pas très pratique quand on doit mettre à jour son site plusieurs fois dans la même journée...  
Un site statique est adapté pour un **site « vitrine »** (pour présenter par exemple son entreprise), mais sans aller plus loin.



Ce type de site se fait de plus en plus rare aujourd'hui, car dès que l'on rajoute un élément d'interaction (comme un **formulaire de contact**), on ne parle plus de site statique mais de site dynamique.

### 1.2.1.2. Découvrez le principe d'un site dynamique

Plus complexe, un site dynamique utilise d'autres langages en plus de HTML et CSS, tels que PHP<sup>2</sup> et MySQL<sup>3</sup>.

Le contenu de ce type de site est dit « dynamique » parce qu'il peut changer sans l'intervention du webmaster !

## 1.2.2. Comprenez le fonctionnement d'un site web

Lorsque vous voulez visiter un site web, vous tapez son adresse dans votre navigateur web. Mais ne vous êtes-vous jamais demandé comment faisait la page web pour arriver jusqu'à vous ?

Il faut savoir qu'Internet est un réseau composé d'ordinateurs. Ceux-ci peuvent être classés en deux catégories :

1. Les **clients** : ce sont les ordinateurs des internautes comme vous. Votre ordinateur fait donc partie de la catégorie des clients. Chaque client représente un visiteur d'un site web.
2. Les **serveurs** : ce sont des ordinateurs puissants qui stockent et délivrent des sites web aux internautes, c'est-à-dire aux clients. La plupart des internautes n'ont jamais vu un serveur de leur vie. Pourtant, les serveurs sont indispensables au bon fonctionnement du Web.

<sup>2</sup> <https://fr.wikipedia.org/wiki/PHP>

<sup>3</sup> <https://fr.wikipedia.org/wiki/MySQL>

	<h1>Concevoir un site web avec PHP et MySQL</h1>	Réalisé le :	28-12-2024
Nombre de pages : 61		Modifié le :	28/12/2024



Vous avez envie d'en savoir plus le modèle client-serveur ? N'hésitez pas à suivre le cours [comprendre le Web](https://openclassrooms.com/fr/courses/1946386-comprendre-le-web)<sup>4</sup>.

Comment les deux communiquent-ils ?

C'est justement là que se fait la différence entre un site statique et un site dynamique. Voyons ensemble ce qui change.

### 1.2.2.1. Consultez un site statique

Lorsque vous vous rendez sur site statique, c'est très simple. Cela se passe en deux temps :

1. Le client demande au serveur à voir une page web.
2. Le serveur lui répond en lui envoyant la page réclamée.

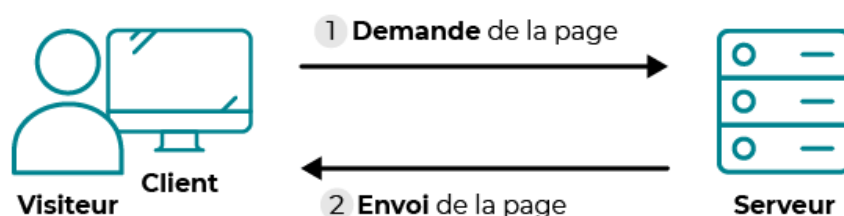


Figure 1: Transfert avec un site statique

La communication est donc plutôt basique :

- « Bonjour, je suis le client, je voudrais voir cette page web. »
- « Tiens, voilà la page que tu m'as demandée. »

Sur un site statique, il ne se passe rien d'autre. Le serveur stocke des pages web et les envoie aux clients qui les demandent, **sans les modifier**. 🖱️

### 1.2.2.2. Consultez un site dynamique

Lorsque vous consultez un site dynamique, il y a une étape intermédiaire : la page est **générée**. 🖱️

Qu'est-ce que ça veut dire ?

Eh bien, il y a une étape supplémentaire, et elle se situe entre les deux étapes de base :

1. Le client demande au serveur à voir une page web.
2. 🖱️ **Le serveur prépare la page** spécialement pour le client (il la génère).
3. Le serveur lui envoie la page qu'il vient de générer.

<sup>4</sup> <https://openclassrooms.com/fr/courses/1946386-comprendre-le-web>

	<h1>Concevoir un site web avec PHP et MySQL</h1>	Réalisé le :	28-12-2024
Nombre de pages : 61		Modifié le :	28/12/2024

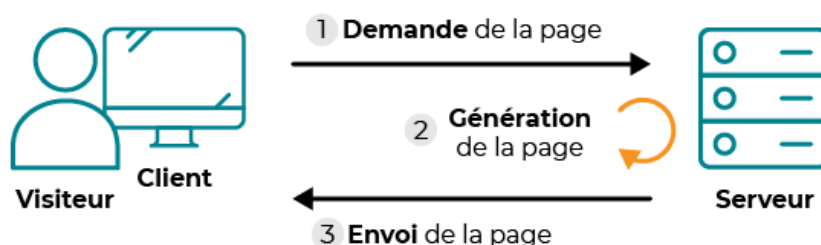


Figure 2: Transfert avec un site dynamique

La page web est générée à chaque fois qu'un client la réclame. C'est précisément ce qui rend les sites dynamiques "vivants" : le contenu d'une même page peut changer d'un instant à l'autre.



C'est comme cela que certains sites parviennent à afficher par exemple votre **pseudonyme** sur toutes les pages. Étant donné que le serveur génère une page à chaque fois qu'on lui en demande une, il peut la personnaliser en fonction des goûts et des préférences du visiteur.

### 1.2.3. Exploitez les langages du Web

Lorsqu'on crée un site web, on est amené à manipuler non pas un, mais plusieurs langages. En tant que webmaster, il faut impérativement les connaître.

#### 1.2.3.1. Utilisez HTML et CSS pour un site statique

De nombreux langages ont été créés pour produire des sites web. Deux d'entre eux constituent une base incontournable pour tous les webmasters.

##### 1.2.3.1.1. HTML

C'est le langage à la base des sites web. Simple à apprendre, il fonctionne à partir de balises. Voici un exemple de code HTML :

```
<p>Bonjour, je suis un <em>paragraphe</em> de texte !</p>
```

##### 1.2.3.1.2. CSS

C'est le langage de mise en forme des sites web. Alors que le HTML permet d'écrire le contenu de vos pages web et de le structurer, le langage CSS s'occupe de la mise en forme et de la mise en page.

C'est en CSS que l'on choisit notamment la couleur, la taille des menus et bien d'autres choses encore.

Voici un code CSS :

```
div.banner {
    text-align: center;
    font-weight: bold;
    font-size: 120%;
}
```

	<h1>Concevoir un site web avec PHP et MySQL</h1>	Réalisé le :	28-12-2024
Nombre de pages : 61		Modifié le :	28/12/2024

}



Ces langages sont la base de tous les sites web. Lorsque le serveur envoie la page web au client, il envoie en fait du code écrit avec les langages HTML et CSS.



Le problème, c'est que lorsqu'on connaît seulement HTML et CSS, on ne peut produire que des sites statiques... et non des sites dynamiques ! Pour ces derniers, il est nécessaire de manipuler d'autres langages, en plus de HTML et CSS.

#### 1.2.3.2. Ajoutez PHP et MySQL pour un site dynamique

Quel que soit le site web que l'on souhaite créer, HTML et CSS sont donc indispensables. Cependant, ils ne suffisent pas pour réaliser des sites dynamiques. Il faut les compléter avec d'autres langages.

C'est justement tout l'objet de ce cours : vous allez apprendre à manipuler PHP et MySQL pour réaliser un site web dynamique<sup>5</sup>.

##### 1.2.3.2.1. PHP

C'est un langage que seuls les serveurs comprennent, et qui permet de rendre votre site dynamique. C'est PHP qui « génère » la page web comme on l'a vu sur un des schémas précédents.

Ce sera le premier langage que nous découvrirons dans ce cours.

Voici un code PHP :

```
<?php echo "Vous êtes le visiteur n°" . $nbre_visiteurs; ?>
```



Le langage PHP peut fonctionner seul, mais il ne prend vraiment de l'intérêt que s'il est combiné à un outil tel que MySQL.

##### 1.2.3.2.2. MySQL

MySQL est ce que l'on appelle un Système de Gestion de Bases de Données <sup>6</sup>(SGBD).

Pour faire simple : son rôle est d'enregistrer des données de manière organisée, afin de vous aider à les retrouver facilement plus tard.

C'est grâce à MySQL que vous pourrez enregistrer :

- la liste des membres de votre site web ;
- les messages postés sur le forum ;
- etc.

##### 1.2.3.2.3. SQL

Le langage qui permet de communiquer avec la base de données s'appelle le SQL<sup>7</sup>.

<sup>5</sup> Quid de JavaScript ?

<sup>6</sup> Plus exactement : Système de Gestion de Bases de Données Relationnel (SGDBR)

<sup>7</sup> [https://fr.wikipedia.org/wiki/Structured\\_Query\\_Language](https://fr.wikipedia.org/wiki/Structured_Query_Language)



	<h1>Concevoir un site web avec PHP et MySQL</h1>	Réalisé le :	28-12-2024
Nombre de pages : 61		Modifié le :	28/12/2024

Voici un code en langage SQL :

```
SELECT id, auteur, message, datemsg FROM livreor ORDER BY datemsg DESC LIMIT 0, 10
```



PHP et MySQL sont disponibles **gratuitement et sous licence Open Source**. Cela signifie une chose essentielle : vous n'aurez pas à déboursier un centime<sup>8</sup> ⚠ pour construire votre site web !

Oublions pour le moment MySQL et concentrons-nous sur PHP.

Les **clients**<sup>9</sup> sont incapables de 🖱 comprendre le code PHP : ils ne connaissent que le HTML et le CSS. Seul le **serveur**<sup>10</sup> est capable de 🖱 lire du PHP. Le rôle de PHP est justement de générer du code HTML, code qui est ensuite envoyé au client de la même manière qu'un site statique, comme le montre la figure suivante :

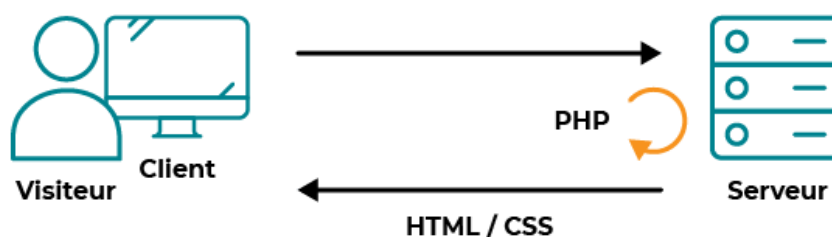


Figure 3: PHP décide ce qui va être affiché sur la page web envoyée au visiteur

PHP est un langage de programmation utilisé sur de nombreux serveurs pour prendre des décisions. C'est PHP qui décide du code HTML qui sera généré et envoyé au client à chaque fois. Pour bien comprendre l'intérêt de tout cela, prenons un exemple.

On peut écrire en PHP :

« Si le visiteur est membre de mon site et qu'il s'appelle Jonathan, affiche **Bienvenue Jonathan** sur la page web. En revanche, si ce n'est pas un membre de mon site, affiche **Bienvenue** à la place, et propose au visiteur de s'inscrire. »

C'est un exemple très basique de site dynamique : selon que vous êtes un membre enregistré ou non, vous **ne verrez pas les mêmes choses**, et **n'aurez peut-être pas accès au même contenu**. 😞

<sup>8</sup> À condition de disposer d'un ordinateur avec un OS et d'une connexion à internet.

<sup>9</sup> Au sens ordinateur (Car c'est une prosopopée).

<sup>10</sup> Encore une prosopopée (On ne les relèvera plus...)

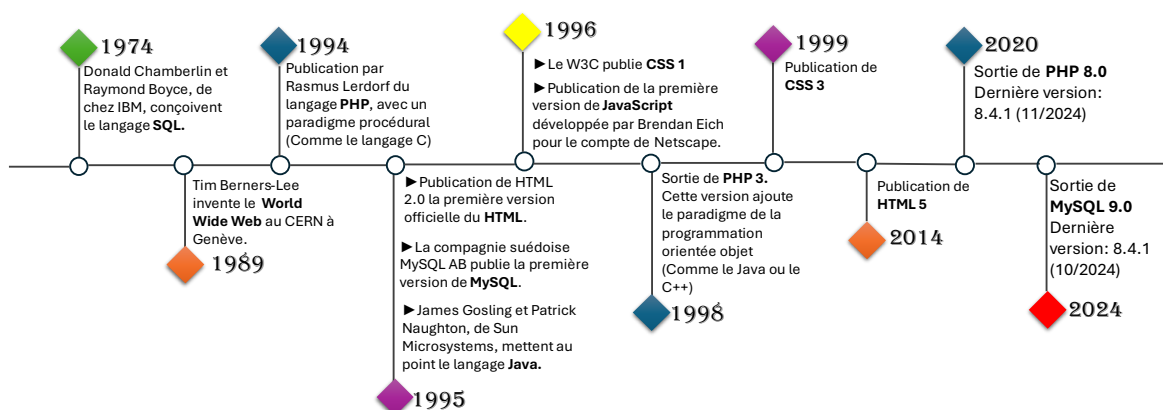
	<h1>Concevoir un site web avec PHP et MySQL</h1>	Réalisé le :	28-12-2024
Nombre de pages : 61		Modifié le :	28/12/2024

#### 1.2.4. En résumé

- Il existe deux types de sites web :
  1. Les sites **statiques** : réalisés en HTML et CSS, leur contenu ne peut être mis à jour que par le webmaster.
  2. Les sites **dynamiques** : réalisés avec d'autres outils comme PHP et MySQL en plus de HTML et CSS, ils permettent aux visiteurs de participer à la vie du site, de poster des messages... bref, de rendre le site vivant !
- Les visiteurs du site sont appelés les "**clients**". Ils demandent au **serveur** qui héberge le site de leur transmettre les pages web.
- PHP est un langage exécuté par le serveur. Il permet de personnaliser la page en fonction du visiteur, de traiter ses messages, d'effectuer des calculs, etc. Il génère une page HTML.
- MySQL est un système de gestion de bases de données. Il se charge du stockage des informations (liste des messages, des membres...).

#### 1.2.5. Timeline des principales technologies du web

Timeline des principales technologies du web 1974→2024 (50 ans)



### 1.3. Préparer l'environnement de travail

Nous savons désormais que PHP s'exécute sur le serveur, et que son rôle est de générer des pages web. Cependant, seul un serveur peut lire du PHP ; or, votre ordinateur n'est pas un serveur.

	<h1>Concevoir un site web avec PHP et MySQL</h1>	Réalisé le :	28-12-2024
Nombre de pages : 61		Modifié le :	28/12/2024

Qu'à cela ne tienne : nous allons temporairement transformer votre ordinateur en serveur pour que vous puissiez exécuter du PHP et travailler sur votre site dynamique.

### 1.3.1. Ayez en tête les outils de base pour créer un site statique

Les webmasters qui créent des sites statiques avec HTML et CSS ont de la chance, ils ont en général déjà tous les programmes dont ils ont besoin :

1. Un éditeur de texte.
2. Un navigateur web.

#### 1.3.1.1. Un éditeur de texte

En théorie, un programme tel que le bloc-notes livré avec Windows suffit, bien qu'il soit recommandé d'utiliser un outil un peu plus évolué, comme Notepad++.

#### 1.3.1.2. Un navigateur web

Il permet de tester la page web.



Il est d'ailleurs conseillé de tester son site régulièrement sur différents navigateurs.

Pour créer un site dynamique, ces outils ne suffisent pas...

Il est nécessaire d'installer des programmes supplémentaires !

### 1.3.2. Utilisez les outils pour créer un site dynamique

Pour que votre ordinateur puisse lire du PHP, il faut qu'il se comporte comme un serveur.

Il suffit simplement d'installer les mêmes programmes que ceux que l'on trouve sur les serveurs qui délivrent les sites web aux internautes.

#### 1.3.2.1. Apache

👉 C'est ce qu'on appelle un **serveur web**. Il s'agit du plus important de tous les programmes, car c'est lui qui est chargé de délivrer les pages web aux visiteurs.

👉 Cependant, **Apache ne gère que les sites web statiques** (il ne peut traiter que des pages HTML). Il faut donc le compléter avec d'autres programmes.

#### 1.3.2.2. PHP

C'est un plug-in pour Apache qui le rend capable de traiter des pages web dynamiques en PHP.

👉 En clair, **en combinant Apache et PHP**, notre ordinateur sera capable de lire des pages web en PHP.

#### 1.3.2.3. MySQL

Il permet d'enregistrer des données de manière organisée (comme la liste des membres de votre site).


Tous ces éléments sont libres et gratuits.

	<h1>Concevoir un site web avec PHP et MySQL</h1>	Réalisé le :	28-12-2024
Nombre de pages : 61		Modifié le :	28/12/2024

Je vous propose d'utiliser **XAMPP** qui marche à la fois pour Windows, pour Mac et pour Linux. Il a l'avantage d'être régulièrement mis à jour.

### 1.3.3. Installez XAMPP sous Windows



Il existe aussi WAMP et MAMP pour Windows. Vous pouvez les essayer,  **mais MAMP n'est pas souvent mis à jour sur Windows**. Je vous recommande donc d'essayer XAMPP en premier.

#### 1.3.3.1. Installation de Xampp

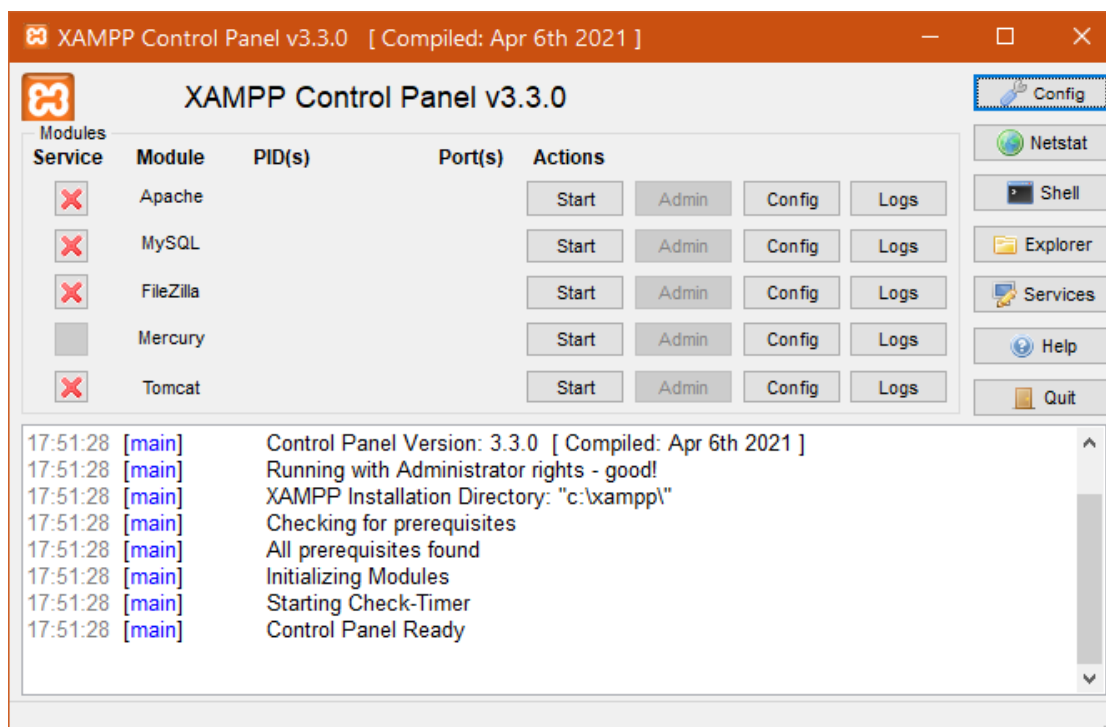


Commencez par télécharger [XAMPP sur son site](https://www.apachefriends.org/fr/index.html)<sup>11</sup>. Prenez la version correspondante à votre système d'exploitation. Une fois téléchargé, installez-le.

Lancez ensuite XAMPP. La fenêtre suivante devrait apparaître :

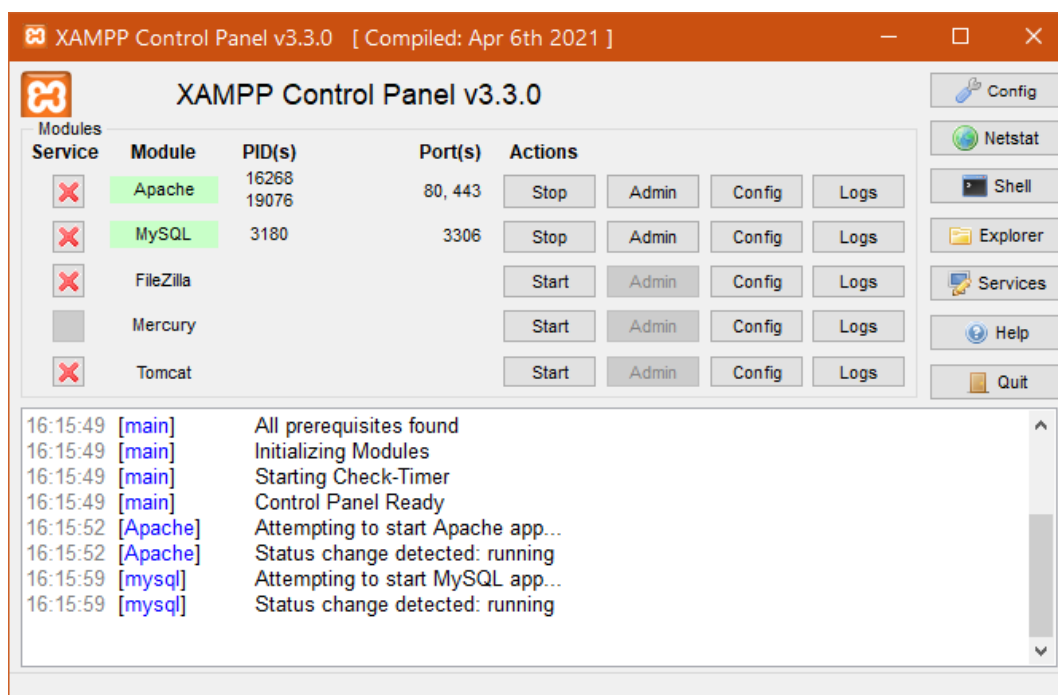
<sup>11</sup> <https://www.apachefriends.org/fr/index.html>

	<h1>Concevoir un site web avec PHP et MySQL</h1>	Réalisé le :	28-12-2024
Nombre de pages : 61		Modifié le :	28/12/2024



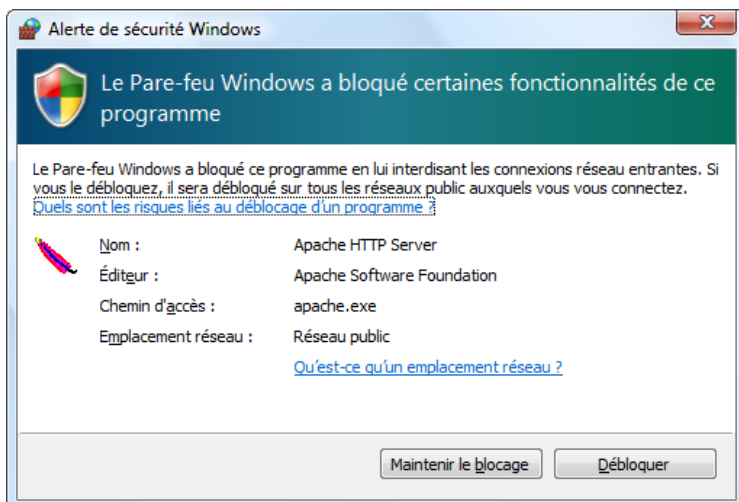
Après le démarrage de XAMPP, vous devez lancer deux programmes essentiels : Apache et MySQL. Cliquez sur le bouton "Start" pour Apache et sur le bouton "Start" pour MySQL.

Vous devriez observer un fond de couleur verte apparaître pour Apache et MySQL. Il peut parfois s'écouler un court laps de temps avant que ces programmes ne soient pleinement opérationnels.



	<h1>Concevoir un site web avec PHP et MySQL</h1>	Réalisé le :	28-12-2024
Nombre de pages : 61		Modifié le :	28/12/2024

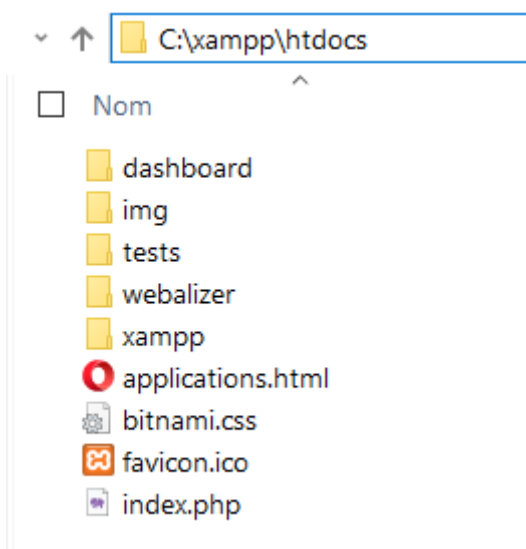
Si une fenêtre apparaît pour vous indiquer que le pare-feu bloque Apache ou MySQL, cliquez sur "Débloquer". Vous n'avez aucune raison de vous inquiéter, c'est parfaitement normal.



Vérifions maintenant que XAMPP est bien installé. Je vous propose de créer un projet de test que nous appellerons tests.

### 1.3.3.2. Test de l'installation de Xampp

Pour ce faire cliquez sur le bouton "Explorer" de XAMPP Control Panel et rendez-vous dans le dossier racine `htdocs`. Puis, créez un nouveau sous-dossier que vous appellerez `tests`, comme sur cette image :

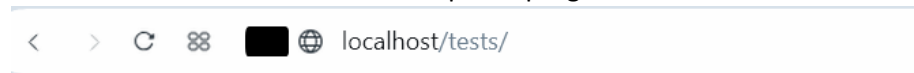


Maintenant ouvrez votre navigateur web. Dans la barre d'adresse, saisissez `localhost/tests`. Le dossier `tests` apparaît dans le navigateur.

Il n'y a rien à l'intérieur pour le moment :

	<h1>Concevoir un site web avec PHP et MySQL</h1>	Réalisé le :	28-12-2024
Nombre de pages : 61		Modifié le :	28/12/2024

Si vous avez le même résultat, cela signifie que tout fonctionne. Bravo, vous avez installé XAMPP et il fonctionne correctement. Vous êtes prêt à programmer en PHP !



## Index of /tests

Name	Last modified	Size	Description
 <a href="#">Parent Directory</a>	-		

Apache/2.4.58 (Win64) OpenSSL/3.1.3 PHP/8.2.12 Server at localhost Port 80

Si vous avez le même résultat, cela signifie que tout fonctionne. Bravo, vous avez installé XAMPP et il fonctionne correctement. Vous êtes prêt à programmer en PHP !

### 1.3.4. Installez XAMPP sous macOS

Vous pouvez passer la section suivante qui ne concerne que les utilisateurs sous Mac.

*1.3.4.1. Ne concerne que les utilisateurs sous Mac* ❌

### 1.3.5. Installez XAMPP sous Linux ★



Sous Linux, il est courant d'installer Apache, PHP et MySQL séparément. Toutefois, il existe aussi des packs tout prêts comme XAMPP (pour X, Apache, MySQL, Perl, PHP), anciennement connu sous le nom de LAMP.

Ce pack est plus complet que MAMP. Nous n'utiliserons toutefois qu'une partie des éléments installés.

*1.3.5.1. Installation de Xampp*

	<h1>Concevoir un site web avec PHP et MySQL</h1>	Réalisé le :	28-12-2024
Nombre de pages : 61		Modifié le :	28/12/2024



Une fois le téléchargement terminé, ouvrez une console. L'installation et le lancement de XAMPP se font en effet uniquement en console.

Rendez-vous dans le dossier dans lequel vous avez téléchargé XAMPP. Par exemple, dans mon cas, le fichier se trouve sur le bureau :

```
cd /Desktop
```

👉 Vous devez passer `root` pour installer et lancer XAMPP.



`root` est le compte administrateur de la machine, qui a notamment le droit d'installer des programmes.

Normalement, il suffit de taper `su` et de rentrer le mot de passe `root`.

👉 Sous Ubuntu, il faudra taper `sudo su` et taper votre mot de passe habituel.

Si comme moi vous utilisez Ubuntu, tapez donc :

```
sudo su
```

Donnez les droits d'exécution au fichier que vous venez de télécharger :

```
chmod 755 xampp-linux-*-installer.run
```

Puis lancez le programme d'installation :

```
./xampp-linux-*-installer.run
```

👉 Pensez à **remplacer l'étoile** dans la commande par le numéro de version du fichier téléchargé.



	<h1>Concevoir un site web avec PHP et MySQL</h1>	Réalisé le :	28-12-2024
Nombre de pages : 61		Modifié le :	28/12/2024

Et voilà ! XAMPP est maintenant installé.

### 1.3.5.2. Démarrage/ Arrêt de Xampp

Pour démarrer XAMPP (et donc Apache, PHP et MySQL), tapez la commande suivante :

```
/opt/lampp/lampp start
```

Si vous désirez plus tard arrêter XAMPP, tapez :

```
/opt/lampp/lampp stop
```

👉 N'oubliez pas que vous devez être root lorsque vous démarrez ou arrêtez XAMPP.

### 1.3.5.3. Test de Xampp

Vous pouvez maintenant tester XAMPP en ouvrant votre navigateur favori et en tapant l'adresse suivante : `http://localhost` .

Les fichiers PHP devront être placés dans le répertoire `/opt/lampp/htdocs` . Vous pouvez y créer un sous-répertoire tests pour vos premiers tests.

```
cd /opt/lampp/htdocs && mkdir tests
```

Une fois le dossier créé, vous pouvez y accéder depuis votre navigateur à l'adresse suivante

```
http://localhost/tests
```

Si vous voyez un dossier vide dans votre navigateur, XAMPP est bien fonctionnel !

### 1.3.6. Découvrez le serveur PHP intégré

Nous avons installé des logiciels qui reproduisent le comportement exact d'un serveur tel qu'il serait configuré et installé en ligne.

Mais pour de petits travaux sur votre ordinateur "en local", PHP fournit un serveur web interne très pratique, et qui utilise PHP en ligne de commande pour provoquer l'exécution du script et le rendu de la page.

Par exemple, si l'on crée un fichier PHP `index.php` avec le contenu suivant :

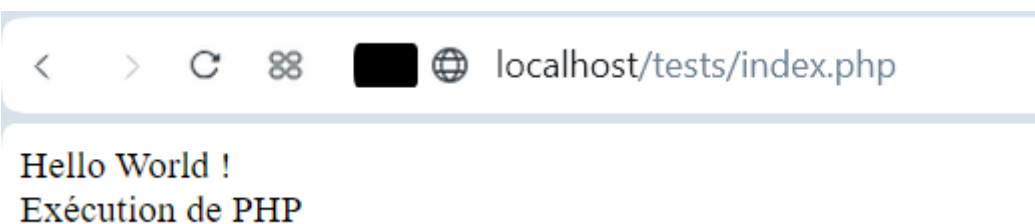
#### 1.3.6.1. Code PHP

```
<?php
echo "Hello World ! ";
echo "<br>" ;
echo "Exécution de PHP";
```

Ensuite, en accédant à : `http://localhost/tests/index.php`

On obtient :

	<h1>Concevoir un site web avec PHP et MySQL</h1>	Réalisé le :	28-12-2024
Nombre de pages : 61		Modifié le :	28/12/2024



### 1.3.7. Utiliser un bon éditeur de texte

Comme vous devez déjà le savoir, pour éditer le code d'une page web, vous avez plusieurs solutions :

- utiliser un éditeur de texte tout simple que vous avez déjà, comme Bloc-notes. Ce logiciel suffit normalement à écrire des pages web en HTML et même en PHP, mais...
- le mieux reste d'utiliser un logiciel spécialisé qui colore votre code (très pratique) et qui numérote vos lignes (très pratique aussi).

Il existe des centaines et des centaines de logiciels gratuits faits pour les développeurs comme vous.

Je vais vous en présenter ici deux :

1. Un que l'on peut utiliser gratuitement : Visual Studio Code.
2. Et un payant : PHPStorm.

Je vous propose donc d'installer un logiciel qui va vous permettre d'éditer vos fichiers sources de manière efficace. Vous en avez probablement déjà installé un si vous avez appris à programmer en HTML/CSS, mais comme on n'est jamais trop prudent, je vais rapidement vous en présenter quelques-uns en fonction de votre système d'exploitation.

Voici le code source HTML que nous allons utiliser pour commencer en terrain connu. Copiez-collez ce code dans l'éditeur de texte que je vais vous faire installer :

#### 1.3.7.1. Code HTML

```
<html>
  <head>
    <meta charset="utf-8"/>
    <title>Ceci est une page HTML de test</title>
  </head>

  <body>
    <h2>Page de test</h2>
    <p>
      Cette page contient <strong>uniquement</strong> du code HTML.<br/>

      Voici quelques petits tests :
    </p>

    <ul>
      <li style="color: blue;">Texte en bleu</li>
      <li style="color: red;">Texte en rouge</li>
      <li style="color: green;">Texte en vert</li>
    </ul>
  </body>
</html>
```

	<h1>Concevoir un site web avec PHP et MySQL</h1>	Réalisé le :	28-12-2024
Nombre de pages : 61		Modifié le :	28/12/2024

Il n'y a pas de PHP pour l'instant, afin de commencer en douceur. Nous allons simplement essayer d'enregistrer un fichier HTML avec ce code pour nous échauffer.

### 1.3.7.2. Visual Studio Code

Que vous soyez sous Windows, Mac ou Linux, je vous recommande de commencer avec l'éditeur [Visual Studio Code](https://code.visualstudio.com/)<sup>12</sup> qui est suffisamment léger et simple si vous débutez.

Ne vous fiez pas à son apparente simplicité : Visual Studio Code est en effet rapide et simple à la base, mais il est possible d'étendre ses fonctionnalités avec d'innombrables plugins !



Visual Studio Code est un très bon éditeur, utilisé par de nombreux développeurs (y compris des professionnels). Il voit en revanche ses limites sur de gros projets, où certains lui préfèrent PHPStorm.

### 1.3.7.3. PHPStorm



[PHPStorm](https://www.jetbrains.com/phpstorm/)<sup>13</sup> ressemble un peu plus à une "machine de guerre". Et pour cause : c'est un IDE, un environnement de travail de développeur. Il est utilisé par de nombreux développeurs PHP professionnels de ma connaissance.

<sup>12</sup> <https://code.visualstudio.com/>

<sup>13</sup> <https://www.jetbrains.com/phpstorm/>

	<h1>Concevoir un site web avec PHP et MySQL</h1>	Réalisé le :	28-12-2024
Nombre de pages : 61		Modifié le :	28/12/2024

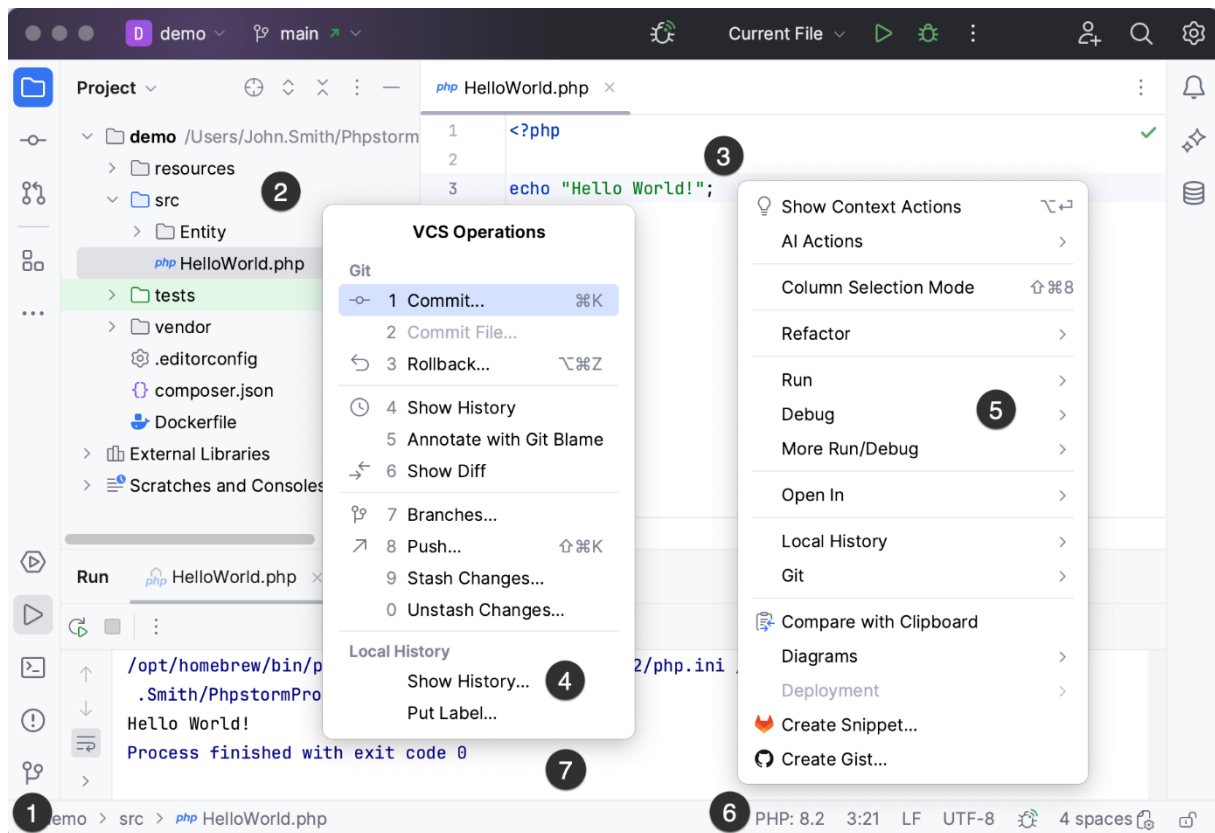


Figure 4: PhpStorm User interface

1. Navigation bar
2. Project tool window
3. Editor
4. Popup menu
5. Context menu
6. Status bar
7. Run tool window<sup>14</sup>



PHPStorm est plus "costaud" que Visual Studio Code. Il met plus de temps à charger, il peut avoir de nombreuses fonctionnalités avancées grâce à ses plugins.

Par ailleurs, PHPStorm est  **payant**.

Vous ne commencerez peut-être pas de suite avec PHPStorm, mais gardez-le sur votre radar car c'est un outil très utilisé que **vous essaieriez sûrement un jour** 👍.

<sup>14</sup> <https://www.jetbrains.com/help/phpstorm/guided-tour-around-the-user-interface.html>

	<h1>Concevoir un site web avec PHP et MySQL</h1>	Réalisé le :	28-12-2024
Nombre de pages : 61		Modifié le :	28/12/2024

### 1.3.8. En résumé

- Pour créer des sites web dynamiques, nous devons installer des outils qui transforment notre ordinateur en serveur, afin de pouvoir tester notre site.
- Les principaux outils nécessaires sont :
  1. Apache, le serveur web.
  2. PHP, le programme qui permet au serveur web d'exécuter des pages PHP.
  3. MySQL, le logiciel de gestion de bases de données.
- Bien qu'il soit possible d'installer ces outils séparément, il est plus simple pour nous d'installer un paquetage tout prêt : XAMPP.
- Il est conseillé d'utiliser un éditeur de texte qui colore le code source, comme Visual Studio Code, pour programmer convenablement en PHP.
- **Pour les personnes plus expérimentées qui travaillent sur de gros projets, je recommande PHPStorm.**

## 1.4. Ecrire le premier script

Maintenant, nous allons passer à la pratique et réaliser notre toute première page web en PHP !

Ne vous attendez pas à un résultat extraordinaire, mais cela va vous permettre de prendre vos marques.

Vous allez en particulier comprendre **comment on sépare le code HTML classique du code PHP**.

### 1.4.1. Utilisez des balises PHP

Vous savez que le code source d'une page HTML est constitué de **balises** (aussi appelées **tags**, en anglais). Par exemple, `<ul>` est une balise.

**Le code PHP viendra s'insérer au milieu du code HTML.** On va progressivement placer dans nos pages web des morceaux de **code PHP à l'intérieur du HTML**.

**Ces bouts de code PHP seront les parties dynamiques de la page**, c'est-à-dire les parties qui peuvent changer toutes seules.

Le code suivant illustre cela :

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Ma page web</title>
  </head>
  <body>
    <h1>Ma page web</h1>
    <p>
      Bonjour <!-- Insérer le pseudo du visiteur ici --> !
    </p>
  </body>
</html>
```

	<h1>Concevoir un site web avec PHP et MySQL</h1>	Réalisé le :	28-12-2024
Nombre de pages : 61		Modifié le :	28/12/2024

Comme vous pouvez le voir, on retrouve le code HTML que l'on connaît bien... et on insère en plus des données dynamiques au milieu :

```
<p>
    Bonjour <!-- Insérer le pseudo du visiteur ici --> !
</p>
```

Ici, par exemple, c'est le pseudonyme : il change en fonction du visiteur.

#### 1.4.1.1. Reconnaître la forme d'une balise PHP

Pour utiliser du PHP, on va devoir introduire une nouvelle balise... et celle-ci est un peu spéciale.

1. Elle commence par `<?php`

2. Et se termine par `?>`

C'est à l'intérieur que l'on mettra du code PHP (ce que je vais vous apprendre tout au long de ce cours).

Voici une balise PHP vide :

```
<?php ?>
```

À l'intérieur, on écrira donc du code source PHP :

```
<?php /* Le code PHP se met ici */ ?>
```

On peut sans problème écrire la balise PHP sur **plusieurs lignes**.

En fait, c'est même indispensable car la plupart du temps, **le code PHP fera plusieurs lignes**.

Cela donnera quelque chose comme :

```
<?php
/* Le code PHP se met ici
Et ici
Et encore ici */
?>
```

##### 1.4.1.1.1. Il existe d'autres balises pour utiliser du PHP

Il existe trois autres balises<sup>15</sup> pour utiliser du PHP ; par exemple :

- `<? ?>`
- `<% %>`
- `<?= ?>`

Ne soyez donc pas étonné si vous en voyez.

Néanmoins, `<?php ?>` est la forme la plus correcte ; vous apprendrez donc à vous servir de cette balise et non pas des autres.

<sup>15</sup> Donc cela fait, au total, 4 syntaxes de balises PHP valides.

	<h1>Concevoir un site web avec PHP et MySQL</h1>	Réalisé le :	28-12-2024
Nombre de pages : 61		Modifié le :	28/12/2024

### 1.4.1.2. Insérer une balise PHP au milieu du code HTML

La balise PHP que nous venons de découvrir s'insère au milieu du code HTML, comme je vous l'ai dit plus tôt. Pour reprendre l'exemple que l'on a vu au chapitre précédent :

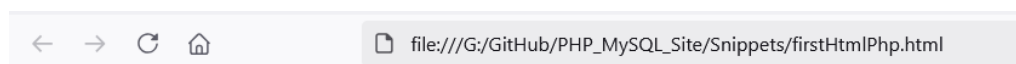
```
<!DOCTYPE html>
<html>
  <head>
    <title>Ceci est une page de test avec des balises PHP</title>
    <meta charset="utf-8" />
  </head>
  <body>
    <h1>Page de test</h1>

    <p>
      Cette page contient du code HTML avec des balises PHP.<br />
      <?php /* Insérer du code PHP ici */ ?>
      Voici quelques petits tests :
    </p>

    <ul>
      <li style="color: blue;">Texte en bleu</li>
      <li style="color: red;">Texte en rouge</li>
      <li style="color: green;">Texte en vert</li>
    </ul>

    <?php
      /* Encore du PHP
      Toujours du PHP */
    ?>
  </body>
</html>
```

Ce qui donne :



## Page de test

Cette page contient du code HTML avec des balises PHP.  
Voici quelques petits tests :

- Texte en bleu
- Texte en rouge
- Texte en vert

### 1.4.1.2.1. On peut placer une balise PHP n'importe où dans le code

Oui ! En effet, on peut vraiment placer du code PHP n'importe où<sup>16</sup> dans la page HTML.

<sup>16</sup> Comme pour le CSS

	<h1>Concevoir un site web avec PHP et MySQL</h1>	Réalisé le :	28-12-2024
Nombre de pages : 61		Modifié le :	28/12/2024

Pas seulement dans le corps de la page, d'ailleurs : vous pouvez placer une **balise PHP dans l'en-tête de la page** (regarder la ligne 4 de l'exemple ci-dessous).

```
<!DOCTYPE html>
<html>
  <head>
    <title>Ceci est une page de test <?php /* Code PHP */ ?></title>
    <meta charset="utf-8" />
  </head>
```

Plus fort encore, vous pouvez même insérer une balise PHP au **milieu d'une balise HTML**, comme le montre la ligne 5 de l'exemple ci-dessous (bon, ce n'est pas très joli, je vous l'accorde) :

```
<!DOCTYPE html>
<html>
  <head>
    <title>Ceci est une page de test</title>
    <meta <?php /* Code PHP */ ?> charset="utf-8" />
  </head>
```

👉 Il faut se rappeler que PHP génère du code HTML. Nous allons mieux comprendre le fonctionnement en apprenant à afficher du texte en PHP.

#### 1.4.2. Affichez du texte



Vous allez un peu mieux comprendre 👉 **comment le PHP fonctionne**, c'est-à-dire **comment il génère du code HTML**.  
Il est indispensable de bien comprendre cela !

##### 1.4.2.1. Utilisez l'instruction *echo*

Le PHP est un langage de programmation, ce qui n'est pas le cas du HTML<sup>17</sup>.

En PHP les instructions se terminent toutes par un point-virgule<sup>18</sup>.



Pour avoir plus d'informations sur l'instruction `echo`, vous pouvez consulter le site officiel de PHP : [l'instruction echo](https://www.php.net/manual/fr/function.echo.php)<sup>19</sup>. Le site officiel de PHP est la principale ressource pour la documentation, les téléchargements et les informations essentielles sur le langage de programmation PHP.

Voici un exemple d'utilisation de cette instruction :

```
<?php echo "Ceci est du texte"; ?>
<!-- Ou bien, avec des parenthèses -->
<?php echo("Ceci est du texte"); ?>
```

<sup>17</sup> HTML (*HyperText Markup Language*) est un langage de balisage.

<sup>18</sup> Comme en C ou en Java, par exemple.

<sup>19</sup> <https://www.php.net/manual/fr/function.echo.php>



	<h1>Concevoir un site web avec PHP et MySQL</h1>	Réalisé le :	28-12-2024
Nombre de pages : 61		Modifié le :	28/12/2024

Comme vous le voyez, à l'intérieur de la balise PHP on écrit l'instruction `echo` suivie du texte à afficher entre guillemets. Les guillemets permettent de délimiter le début et la fin du texte, ce qui aide l'ordinateur à se repérer.

👉 L'instruction se termine par un point-virgule comme je vous l'avais annoncé, ce qui signifie : "Fin de l'instruction".



C'est beaucoup plus rare, mais 👉 l'instruction `echo` peut être entourée de parenthèses ouvrantes et fermantes : nous reviendrons 💡 là-dessus dans le chapitre sur les fonctions.



Notez qu'il existe une instruction identique à `echo` appelée `print`, qui fait la même chose. Cependant, `echo` est plus couramment utilisée. Il faut savoir qu'on a aussi le droit de demander d'afficher des balises. Par exemple, le code suivant fonctionne :

```
<?php echo "Ceci est du <strong>texte</strong>"; ?>
```

Le mot « texte » sera affiché en gras grâce à la présence des balises `<strong>` et `</strong>`.



Comment faire pour afficher un guillemet ?

Si vous mettez un guillemet, ça veut dire pour l'ordinateur que le texte à afficher s'arrête là. Vous risquez au mieux de faire planter votre beau code et d'avoir une terrible « **Parse error** ». !

👉 La solution consiste à faire précéder le guillemet d'un antislash \ :

```
<?php echo "Cette ligne a été écrite \"uniquement\" en PHP."; ?>
```

Vous savez que le code PHP s'insère au milieu du code HTML. Alors allons-y, prenons une page basique en HTML et plaçons-y du code PHP (lignes en gras).

```
<!DOCTYPE html>
<html>
  <head>
    <title>Notre première instruction : echo</title>
    <meta charset="utf-8" />
  </head>
  <body>
    <h2>Affichage de texte avec PHP</h2>

    <p>
      Cette ligne a été écrite entièrement en HTML.<br />
      <?php echo("Celle-ci a été écrite entièrement en PHP."); ?>
    </p>
  </body>
</html>
```

	<h1>Concevoir un site web avec PHP et MySQL</h1>	Réalisé le :	28-12-2024
Nombre de pages : 61		Modifié le :	28/12/2024

Je vous propose de copier-coller ce code source dans votre éditeur de texte (Visual Studio Code ou autre... ) et d'enregistrer la page. Nous allons l'essayer et voir ce qu'elle produit comme résultat.

#### 1.4.2.2. Enregistrez une page PHP

Enregistrez la page avec l'extension .php . Par exemple : affichertexte.php

##### 1.4.2.2.1. Fichier affichertexte.php

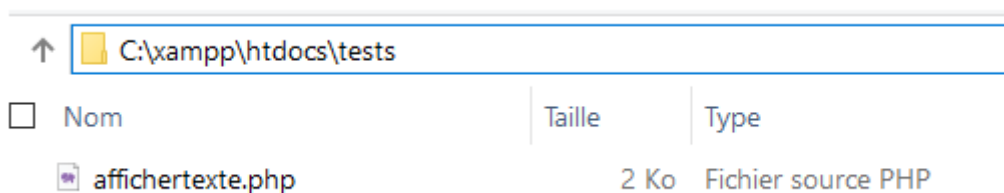
```
<!DOCTYPE html>
<html>
  <head>
    <title>Ceci est une page de test avec des balises PHP</title>
    <meta charset="utf-8" />
  </head>
  <body>
    <h1>Page de test</h1>
    <p>
      Cette page contient du code HTML avec des balises PHP.<br />
      <?php /* Insérer du code PHP ici */ ?>
      Voici quelques petits tests :
    </p>
    <ul>
      <li style="color: blue;">Texte en bleu</li>
      <li style="color: red;">Texte en rouge</li>
      <li style="color: green;">Texte en vert</li>
    </ul>

    <?php
      /* Encore du PHP
      Toujours du PHP */
      ?>
      <?php echo "Ceci est du texte"; ?>

  <!-- Ou bien, avec des parenthèses -->
  <?php echo("Ceci est du texte"); ?>
    <?php echo "Ceci est du <strong>texte</strong>"; ?>
    <p>
      Cette ligne a été écrite entièrement en HTML.<br />
      <?php echo("Celle-ci a été écrite entièrement en PHP."); ?>
    </p>
  </body>
</html>
```


Ce fichier se situe dans le dossier tests que je vous ai fait créer. Il doit se trouver dans C:\XAMPP\htdocs\tests sous Windows,

	<h1>Concevoir un site web avec PHP et MySQL</h1>	Réalisé le :	28-12-2024
Nombre de pages : 61		Modifié le :	28/12/2024



ou /Applications/XAMPP/htdocs/tests sous Mac,  
ou /opt/lampp/htdocs/tests/ sous Linux.

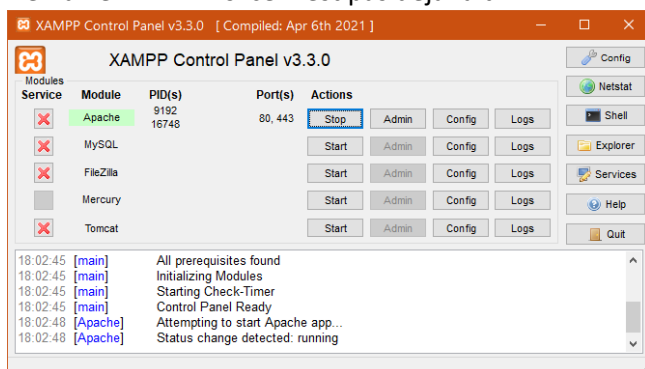


Quel que soit votre système d'exploitation, l'essentiel est que le fichier soit enregistré dans le dossier `www` si vous utilisez  MAMP ou WAMP (ou un de ses sous-dossiers) ; sinon le fichier PHP ne pourra pas s'exécuter !

#### 1.4.2.3. Testez une page PHP

Pour tester votre page PHP, tout dépend de votre système d'exploitation, mais la manœuvre est la même dans les grandes lignes.



1. Démarrez XAMPP si ce n'est pas déjà fait.



2. Allez à l'adresse `http://localhost/tests` . Une page web s'ouvre, indiquant tous les fichiers qui se trouvent dans le dossier `tests` . Vous devriez avoir le fichier `affichertexte.php` .



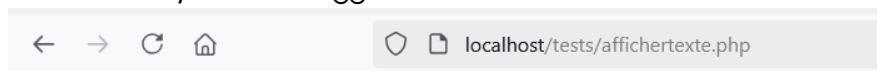
## Index of /tests

<a href="#">Name</a>	<a href="#">Last modified</a>	<a href="#">Size</a>	<a href="#">Description</a>
 <a href="#">Parent Directory</a>		-	
 <a href="#">affichertexte.php</a>	2024-12-16 16:45	1.1K	

Apache/2.4.58 (Win64) OpenSSL/3.1.3 PHP/8.2.12 Server at localhost Port 80

	<h1>Concevoir un site web avec PHP et MySQL</h1>	Réalisé le :	28-12-2024
Nombre de pages : 61		Modifié le :	28/12/2024

3. Cliquez dessus : votre ordinateur génère alors le code PHP puis ouvre la page. Vous avez le résultat devant vos yeux ébahis. 🤖 👁 👁



## Page de test

Cette page contient du code HTML avec des balises PHP.  
Voici quelques petits tests :

- Texte en bleu
- Texte en rouge
- Texte en vert

Ceci est du texte Ceci est du texte Ceci est du **texte**

Cette ligne a été écrite entièrement en HTML.  
Celle-ci a été écrite entièrement en PHP.

Le même résultat peut être obtenu dans votre navigateur, en allant directement à l'adresse suivante <http://localhost/tests/affichertexte.php> .

La méthode devrait être quasiment la même, que vous soyez sous Windows, macOS X ou Linux.



Si vous utilisez le serveur local de PHP, vous pouvez sauvegarder le fichier où vous le souhaitez, mais en contrepartie, vous devez toujours démarrer le serveur au niveau d'un dossier contenant le fichier à exécuter. Pour rappel, voici la commande : `php -S localhost:8080` .

Ne pas oublier : le serveur PHP recherche automatiquement un fichier nommé `index.php` comme 📁 point d'entrée par défaut !



Mais... ce n'est pas plus simple de l'écrire en HTML, finalement ?

Si ! Mais vous verrez bientôt 📁 **l'intérêt de cette fonction**. Pour le moment, on constate juste que ça écrit du texte. 🤖

En attendant, pour vous amuser et comprendre la force de PHP, essayez juste le code suivant (qu'on expliquera plus tard dans le cours) :

```
<!DOCTYPE html>
<html>
  <head>
    <meta charset="utf-8" />
    <title>Ma page web</title>
  </head>
  <body>
    <h1>Ma page web</h1>
```

	<h1>Concevoir un site web avec PHP et MySQL</h1>	Réalisé le :	28-12-2024
Nombre de pages : 61		Modifié le :	28/12/2024

```
<p>Aujourd'hui nous sommes le <?php echo date('d/m/Y h:i:s');
?>.</p>
</body>
</html>
```

Ce qui donne :



## Ma page web

Aujourd'hui nous sommes le 16/12/2024 06:18:13.



Testez ce code et admirez : la date et l'heure s'affichent automatiquement sur la page web !

Attendez quelques minutes, puis actualisez la page : l'heure s'est mise à jour toute seule !



Regardez le code source de la page<sup>20</sup> générée dans le navigateur :

vous verrez **qu'il n'y a pas de code PHP** et que

l'heure a directement été envoyée dans le code HTML après exécution du code PHP par le serveur.



### 1.4.3. Commentez votre code

Un commentaire permet de vous y retrouver dans votre code PHP, parce que si vous n'y touchez pas pendant des semaines et que vous y revenez, vous risquez d'être un peu perdu.

Vous pouvez écrire tout et n'importe quoi, le tout est de s'en servir à bon escient<sup>21</sup>.

<sup>20</sup> Raccourci pour voir le code source de la page: Ctrl+U

<sup>21</sup> Clean code ? (Robert C. Martin 2009)

	<h1>Concevoir un site web avec PHP et MySQL</h1>	Réalisé le :	28-12-2024
Nombre de pages : 61		Modifié le :	28/12/2024

Il existe deux types de commentaire :

1. Les commentaires **monoligne**.
2. Les commentaires **multilignes**.

#### 1.4.3.1. Faites des commentaires monoligne

Pour indiquer que vous écrivez un commentaire sur une seule ligne, vous devez taper deux slashes : « // ». Tapez ensuite votre commentaire<sup>22</sup>.

Un exemple :

```
<?php
echo "J'habite en Chine."; // Cette ligne indique où j'habite

// La ligne suivante indique mon âge
echo "J'ai 92 ans.";
?>
```

Je vous ai mis deux commentaires à des endroits différents :

- le premier est à la fin d'une ligne ;
- le second est sur toute une ligne.

À vous de voir où vous placez vos commentaires : si vous commentez une ligne précise, mieux vaut mettre le commentaire à la fin de cette ligne.

##### 1.4.3.1.1. Commentaire monoligne style Unix

Le signe #<sup>23</sup> permet, également de commenter une ligne en PHP.

```
<?php

# Il est également possible de faire un commentaire sur une ligne

?>
```



Ces commentaires fonctionneront uniquement à l'intérieur des balises PHP, 🖱️ et ils seront exécutés s'ils sont placés dans du code HTML.

#### 1.4.3.2. Faites des commentaires multilignes

Ce sont les plus pratiques si vous pensez écrire un commentaire sur plusieurs lignes, mais on peut aussi s'en servir pour écrire des commentaires d'une seule ligne. Il faut commencer par écrire /\* puis refermer par \*/ :

```
<?php

/* La ligne suivante indique mon âge

Si vous ne me croyez pas...
```

<sup>22</sup> Ceci est un héritage du langage C

<sup>23</sup> Code ascii = 35


	<h1>Concevoir un site web avec PHP et MySQL</h1>	Réalisé le :	28-12-2024
Nombre de pages : 61		Modifié le :	28/12/2024

```
... vous avez raison ;o) */
```


```
echo "J'ai 92 ans.";
```

```
?>
```

#### 1.4.4. En résumé

- Les pages web contenant du PHP ont l'extension `.php`
- **Une page PHP est  une simple page HTML qui contient des instructions en langage PHP.**
- Les instructions PHP sont placées dans une balise ouvrante et fermante : `<?php ?>`
- Pour afficher du texte en PHP, on utilise l'instruction `echo`
- Il est possible d'ajouter des commentaires en PHP pour décrire le fonctionnement du code.  
On utilise pour cela les symboles `//` ou `/* */`

### 1.5. Configurer PHP pour visualiser les erreurs

En effet, lorsqu'un script PHP plante , le comportement par défaut de PHP est de n'afficher qu'une page blanche (une page de navigateur sans contenu).



Pour faciliter notre vie de développeur, **il va falloir faire en sorte que les erreurs PHP s'affichent**. Sinon, nous aurons de grosses difficultés par la suite pour comprendre pourquoi nos pages ne marchent pas.

#### 1.5.1. Configurez PHP pour afficher les erreurs

Eh oui, PHP est configurable ! 



**Si les erreurs s'affichent déjà bien dans votre navigateur, inutile de faire les manipulations qui vont suivre !**

Par défaut, PHP n'affiche pas les erreurs. Pourquoi ?

Pour des raisons de sécurité ! C'est pour éviter de donner trop d'indications aux utilisateurs, tout simplement.



Un mantra à vous répéter :

"Moins l'utilisateur en sait sur mon application, mieux mon application se portera."

La configuration de PHP se fait dans un fichier appelé `php.ini`

##### 1.5.1.1. Localisez le fichier de configuration PHP du serveur web



Pour connaître l'ensemble des informations relatives au PHP utilisé par le serveur web, il existe une commande PHP, `phpinfo()` (on parle de fonction, mais on y reviendra).

 Nombre de pages : 61	<h1>Concevoir un site web avec PHP et MySQL</h1>	Réalisé le :	28-12-2024
		Modifié le :	28/12/2024

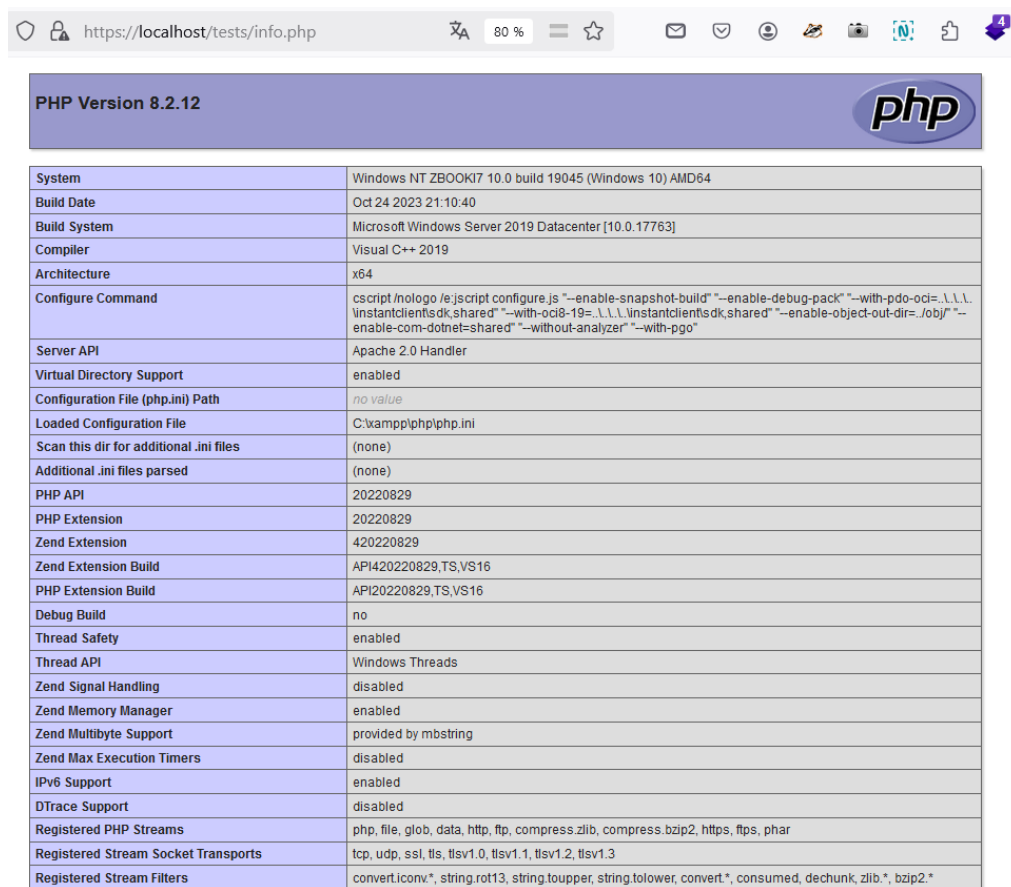
Nous allons utiliser `phpinfo()` pour localiser le fichier de configuration, puis le modifier.

Je vous invite donc à créer un fichier PHP avec le code qui suit :

```
<?php
phpinfo();
?>
```

Enregistrez-le sous le nom `info.php` dans le dossier accessible pour votre serveur web (normalement `htdocs` ou `www`<sup>24</sup> ).

Et enfin, affichez la page. Vous devriez obtenir le résultat suivant :



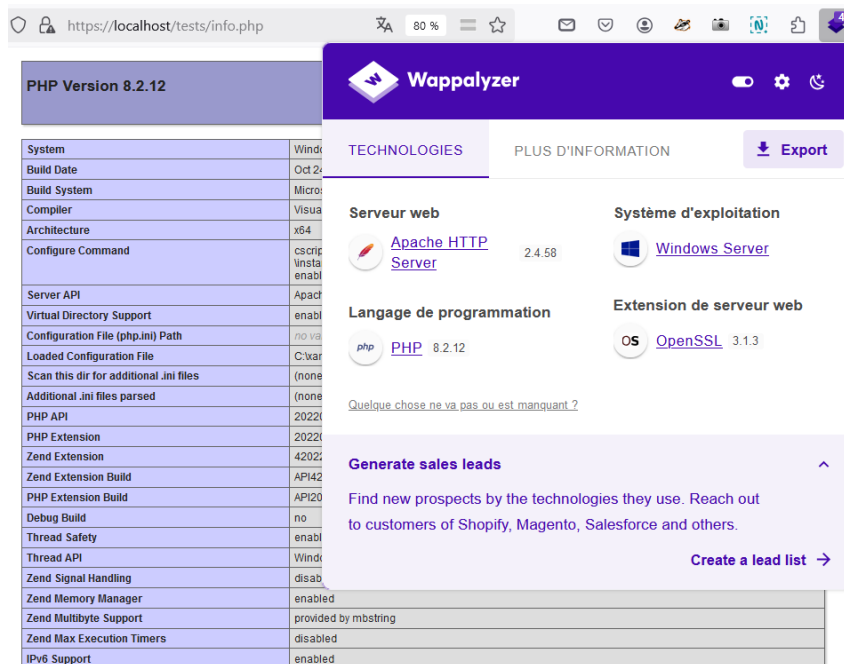
PHP Version 8.2.12	
System	Windows NT ZBOOKI7 10.0 build 19045 (Windows 10) AMD64
Build Date	Oct 24 2023 21:10:40
Build System	Microsoft Windows Server 2019 Datacenter [10.0.17763]
Compiler	Visual C++ 2019
Architecture	x64
Configure Command	cmdscript /nologo /e:jsconfig.js "--enable-snapshot-build" "--enable-debug-pack" "--with-pdo-oci=.\\..\\..\\instantclient\\sdk\\shared" "--with-oci8-19=.\\..\\..\\instantclient\\sdk\\shared" "--enable-object-out-dir=.\\obj\\" "--enable-com-dotnet=shared" "--without-analyzer" "--with-pgo"
Server API	Apache 2.0 Handler
Virtual Directory Support	enabled
Configuration File (php.ini) Path	no value
Loaded Configuration File	C:\xampp\php\php.ini
Scan this dir for additional .ini files	(none)
Additional .ini files parsed	(none)
PHP API	20220829
PHP Extension	20220829
Zend Extension	420220829
Zend Extension Build	API420220829,TS,VS16
PHP Extension Build	API20220829,TS,VS16
Debug Build	no
Thread Safety	enabled
Thread API	Windows Threads
Zend Signal Handling	disabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring
Zend Max Execution Timers	disabled
IPv6 Support	enabled
DTrace Support	disabled
Registered PHP Streams	php, file, glob, data, http, ftp, compress.zlib, compress.bzip2, https, ftps, phar
Registered Stream Socket Transports	tcp, udp, ssl, tls, tlsv1.0, tlsv1.1, tlsv1.2, tlsv1.3
Registered Stream Filters	convert.iconv.*, string.rot13, string.toupper, string.tolower, convert.*, consumed, dechunk, zlib.*, bzip2.*

Figure 5: Résultat du script contenant l'instruction `phpinfo()`

<sup>24</sup> Dans le cas de WAMP



 Nombre de pages : 61	<h1>Concevoir un site web avec PHP et MySQL</h1>	Réalisé le :	28-12-2024
		Modifié le :	28/12/2024



The screenshot shows the Wappalyzer extension interface. On the left, a table lists system and PHP configuration details:

System	Value
Build Date	Oct 21 2022
Build System	Microsoft Visual C++
Compiler	Visual C++
Architecture	x64
Configure Command	configure --enable-zip --enable-gd --enable-mbstring --enable-openssl --enable-sodium --enable-xml --enable-zlib
Server API	Apache2handler
Virtual Directory Support	enabled
Configuration File (php.ini) Path	/usr/local/etc/php/
Loaded Configuration File	C:\xampp\php\php.ini
Scan this dir for additional .ini files	(none)
Additional .ini files parsed	(none)
PHP API	20220829
PHP Extension	20220829
Zend Extension	420220829
Zend Extension Build	API42
PHP Extension Build	API20
Debug Build	no
Thread Safety	enabled
Thread API	Windows
Zend Signal Handling	disabled
Zend Memory Manager	enabled
Zend Multibyte Support	provided by mbstring
Zend Max Execution Timers	disabled
IPv6 Support	enabled

On the right, the Wappalyzer interface displays the following information:

- TECHNOLOGIES**: Apache HTTP Server 2.4.58
- Langage de programmation**: PHP 8.2.12
- Système d'exploitation**: Windows Server
- Extension de serveur web**: OpenSSL 3.1.3

At the bottom, there is a section titled "Generate sales leads" with a link to "Create a lead list".

Figure 6: avec l'extension Wappalyzer

Cette page contient toute une flopée d'informations, dont :

- la version de PHP utilisée (pour moi, il s'agit de PHP 8.2.12) ;
- le type de serveur web (ici Apache) ;
- et la localisation du (ou des fichiers) de configuration pour PHP.

👉 Notez que vous pouvez aussi accéder au `phpinfo` depuis le menu "PHPINFO" en haut de la page d'accueil de XAMPP :



The screenshot shows the XAMPP dashboard. The navigation menu at the top includes links for "Apache Friends", "FAQs", "HOW-TO Guides", "PHPInfo", and "phpMyAdmin". The "PHPInfo" link is highlighted with a yellow cursor. Below the navigation menu, the main heading reads "XAMPP Apache + MariaDB + PHP + Perl".

## Welcome to XAMPP for Windows 8.2.12

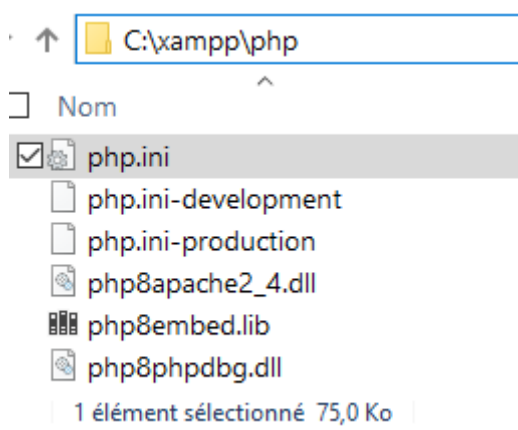
You have successfully installed XAMPP on this system! Now you can start using Apache, MariaDB, PHP and other components. You can find more info in the [FAQs](#) section or check the [HOW-TO Guides](#) for getting started with PHP applications.

 Nombre de pages : 61	<h1>Concevoir un site web avec PHP et MySQL</h1>	Réalisé le :	28-12-2024
		Modifié le :	28/12/2024

Retrouvez la ligne "Loaded Configuration File" (ce qui signifie "fichier de configuration chargé", en anglais), et regardez la valeur. Dans mon cas, on peut voir ceci :



Figure 7: Chemin du fichier de configuration de PHP chargé par le serveur web



On va donc ouvrir le fichier `php.ini`<sup>25</sup> et le modifier.

### 1.5.1.2. Modifiez le fichier de configuration PHP

Il faut s'assurer que :

la clé de configuration `error_reporting` a la valeur `E_ALL`

la clé de configuration `display_errors` a la valeur `On`

Allons-y étape par étape :

1. Effectuez une recherche dans le fichier avec le terme `error_reporting`. S'il n'y a pas écrit `error_reporting = E_ALL`, remplacez-la par la bonne valeur.
2. Ensuite, effectuez une nouvelle recherche dans le fichier avec le terme `display_errors`. S'il n'y a pas écrit `display_errors = On`, remplacez-la par la bonne valeur.
3. **Enregistrez le fichier.**
4. 🖱 Relancez le serveur pour qu'il prenne en compte vos modifications. Il suffit de relancer XAMPP, par exemple.

<sup>25</sup> Il contient 2021 lignes pour 75Ko.

	<h1>Concevoir un site web avec PHP et MySQL</h1>	Réalisé le :	28-12-2024
Nombre de pages : 61		Modifié le :	28/12/2024

```

485 ; Default Value: E_ALL
486 ; Development Value: E_ALL
487 ; Production Value: E_ALL & ~E_DEPRECATED & ~E_STRICT
488 ; https://php.net/error-reporting
489 error_reporting=E_ALL
490
491 ; This directive controls whether or not and where PHP will output errors,
492 ; notices and warnings too. Error output is very useful during development, but
493 ; it could be very dangerous in production environments. Depending on the code
494 ; which is triggering the error, sensitive information could potentially leak
495 ; out of your application such as database usernames and passwords or worse.
496 ; For production environments, we recommend logging errors rather than
497 ; sending them to STDOUT.
498 ; Possible Values:
499 ;   Off = Do not display any errors
500 ;   stderr = Display errors to STDERR (affects only CGI/CLI binaries!)
501 ;   On or stdout = Display errors to STDOUT
502 ; Default Value: On
503 ; Development Value: On
504 ; Production Value: Off
505 ; https://php.net/display-errors
506 display_errors=On
507
508 ; The display of errors which occur during PHP's startup sequence are handled
509 ; separately from display_errors. We strongly recommend you set this to 'off'
510 ; for production environments to avoid leaking information details.

```

Figure 8: Vérifiez que vous avez bien activé les erreurs dans le fichier php.ini



Dans le fichier de configuration `php.ini`, le point-virgule ( ; ) en début de ligne signifie que tout ce qui suit est un commentaire, et est donc ignoré.

Si l'une de ces lignes (ou les deux) sont commentées, il suffit de retirer le point-virgule en début de ligne.



Faites attention à ce que ces lignes de configuration n'existent qu'une seule fois dans le fichier, en effet, ne créez pas ces lignes si elles existaient déjà.

### 1.5.2. Testez l'affichage des erreurs

Nous allons maintenant créer une erreur dans un script PHP pour nous assurer que l'erreur s'affiche dans le navigateur.

Dans le script que nous avons créé pour afficher les informations relatives à PHP pour le serveur web ; nous l'avons appelé `info.php` , retirez une parenthèse, puis enregistrez le fichier.

Ça devrait donner ceci :

```
<?php
phpinfo();
```

	<h1>Concevoir un site web avec PHP et MySQL</h1>	Réalisé le :	28-12-2024
Nombre de pages : 61		Modifié le :	28/12/2024

?>

Maintenant, affichez la page à l'aide de votre navigateur web :

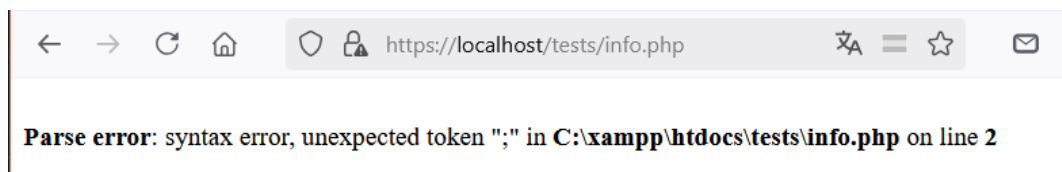


Figure 9: Affichage des erreurs dans le navigateur web

Et voilà !

Si vous voyez bien cette erreur, c'est que PHP est configuré pour afficher le détail des erreurs. Ouf ! Ça nous fera gagner beaucoup de temps pour comprendre nos problèmes par la suite.

👉 Autrement dit cela nous servira à **déboguer** notre code source PHP 👁 👁



26

### 1.5.3. En résumé

- PHP dispose d'un mode de débogage pour afficher les erreurs contenues dans vos scripts.
- Pour activer le débogage, on modifie la configuration de PHP en éditant le fichier **php.ini**.
- On peut trouver la localisation de ce fichier (et plein d'autres informations) en exécutant la fonction **phpinfo()** `<?php phpinfo(); ?>` .
- Pour activer l'affichage, on change la propriété **display\_errors** à **On** et on filtre le type d'erreurs avec **error\_reporting** (on choisira **E\_ALL** pour voir toutes les erreurs !).

*C'est la fin de cette première partie, maintenant testez vos connaissances avec le quiz !*

<sup>26</sup> Image générée par IA

	<h1>Concevoir un site web avec PHP et MySQL</h1>	Réalisé le :	28-12-2024
Nombre de pages : 61		Modifié le :	28/12/2024

## 1.6. Quiz : Premiers pas en PHP

Bravo ! Vous avez réussi cet exercice !

### Compétences évaluées

✓ Installer les outils propres à PHP (serveur web, logiciel de gestion de base de données)

#### Question 1

Quel élément est indispensable au bon fonctionnement d'un site web écrit avec PHP afin qu'il s'affiche correctement dans un navigateur ?

- ✓ ☒ Un serveur web (Apache, Nginx...)
- ☐ Un IDE
- ☐ Un terminal (ligne de commande)

*L'IDE est l'éditeur de texte vous permettant de développer/coder. Cela n'impacte pas la génération de la page.*

*Le terminal (ligne de commande) ne rentre pas en jeu lors de la génération de la page web.*

#### Question 2

PHP permet de créer des sites...

- ☐ statiques
- ✓ ☒ dynamiques

*Les sites dynamiques peuvent changer à chaque visite. Au lieu d'afficher tout le temps la même information, ils peuvent être mis à jour automatiquement et par les visiteurs.*

#### Question 3

MySQL permet...

- ☐ d'afficher des sites
- ☐ de créer des sites dynamiques
- ✓ ☒ de stocker de l'information structurée

*MySQL est ce qu'on appelle un SGBD (système de gestion de bases de données). Son rôle est d'enregistrer des données de manière organisée afin de vous aider à les retrouver facilement plus tard.*

	<h1>Concevoir un site web avec PHP et MySQL</h1>	Réalisé le :	28-12-2024
Nombre de pages : 61		Modifié le :	28/12/2024

## Question 4

**Sélectionnez des éditeurs de texte dans lesquels on peut écrire du code PHP :**

*Attention, plusieurs réponses sont possibles.*

- ☐ Apache
- ✓ ☒ Visual Studio Code
- ✓ ☒ PHPStorm
- ☐ Google Chrome

*Visual Studio Code et PHPStorm sont deux éditeurs connus pour permettre d'écrire du code en PHP, notamment. Apache sert à faire tourner PHP, tandis que Google Chrome est un navigateur web qui sert à afficher les pages web.*

## Question 5

**À quoi ressemble une balise PHP ?**

- ☐ `<? / ?>`
- ☐ `<php ?>`
- ✓ ☒ `<?php ?>`

*Il y a plusieurs façons de créer une balise PHP, mais la plus recommandée et la plus utilisée est* `<?php ?>`

## Question 6

**Pourquoi faut-il configurer PHP pour afficher des erreurs sur sa machine ?**

- ✓ ☒ Pour comprendre nos erreurs et les résoudre plus vite.
- ☐ Pour accélérer le temps d'affichage d'une page.
- ☐ Pour éviter un bug de PHP qui empêche les pages de s'afficher correctement.

*Sans erreur détaillée, impossible de comprendre pourquoi nous avons fait une erreur !  
Configurez bien PHP pour afficher les erreurs, ça va vous être indispensable très vite !*

	<h1>Concevoir un site web avec PHP et MySQL</h1>	Réalisé le :	28-12-2024
Nombre de pages : 61		Modifié le :	28/12/2024

## Question 7

Comment s'appelle le fichier de configuration de PHP ?

- ☐ php.conf  
☒ php.ini  
☐ php\_conf.ini  
☐ php\_ini.conf

Le fichier `php.ini` contient la configuration du langage PHP, c'est dans ce fichier qu'il vous sera possible – par exemple – d'activer le mode de débogage.

## Question 8

Quelles propriétés de la configuration PHP devez-vous modifier pour activer l'affichage des erreurs PHP ?

Attention, plusieurs réponses sont possibles.

- ☐ debug  
☒ error\_reporting  
☐ profiler  
☒ display\_errors

Les propriétés `error_reporting` et `display_errors` sont à modifier pour autoriser l'affichage des erreurs et le type d'erreur que PHP affichera à l'utilisateur.

## 2. Partie 2 - Réalisez un site web dynamique avec PHP

Nous démarrons le projet fil rouge pour développer notre site web dynamique : un site de partage de recettes de cuisine. Et ça commence par une notion fondamentale : les variables.

### 2.1. Décrire les éléments du projet à l'aide de variables

Les variables sont indispensables à tout langage de programmation, et en PHP on n'y échappe pas.

Pour pouvoir développer notre projet de partage, nous allons avoir besoin de structurer l'application autour d'objets qui la composent. C'est ce qu'on appelle des objets "métiers".

 Nombre de pages : 61	<h1>Concevoir un site web avec PHP et MySQL</h1>	Réalisé le :	28-12-2024
		Modifié le :	28/12/2024

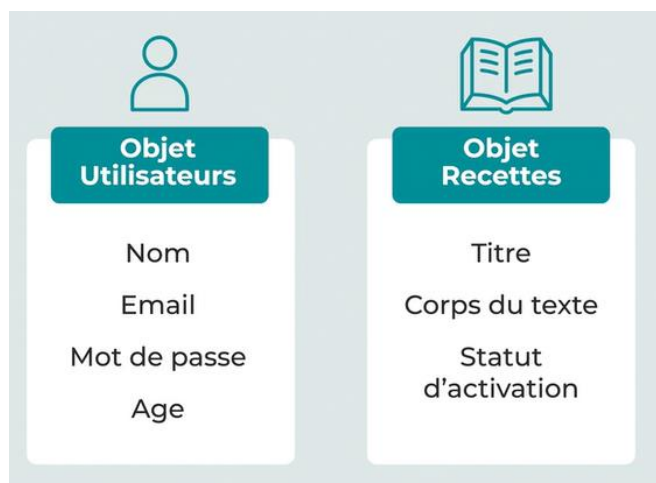


Figure 100: Les objets métiers


Pour un site de partage de recettes de cuisine, c'est simple :

- Des utilisateurs se connectent : ils ont un nom, un e-mail, un mot de passe, un âge...
- Ils consultent ou créent des recettes : elles ont un titre, un corps (la recette), un statut d'activation...
- Et ainsi de suite pour chacun des objets qui constituent votre projet.

### 2.1.1. Comprendre ce qu'est une variable

En effet, le propre d'une variable c'est de pouvoir varier<sup>27</sup>.



Une **variable**, c'est une petite information stockée en  **mémoire vive**<sup>28</sup> **temporairement**<sup>29</sup>. En PHP, la variable (l'information) existe tant que la page est en cours de génération. Dès que la page PHP est générée, toutes les variables sont supprimées de la mémoire vive, car elles ne servent plus à rien.

Ce n'est donc pas un fichier qui reste stocké sur le disque dur, mais une petite information **temporaire présente** en mémoire vive.

C'est à vous de créer des variables. Vous en créez quand vous en avez besoin pour retenir des informations.

#### 2.1.1.1. Donner toujours un nom et une valeur aux variables

Une variable est toujours constituée de deux éléments :

1. **son nom** : pour pouvoir la reconnaître, vous devez donner un nom à votre variable. Par exemple `age` ;
2. **sa valeur** : c'est l'information qu'elle contient, et qui peut changer. Par exemple : `17` .

Ici, je vous ai donné l'exemple d'une variable appelée `age` qui a pour valeur `17` .

<sup>27</sup> C'est une vérité de La Palice (Jacques II de Chabannes). i.e., un truisme.

<sup>28</sup> La RAM

<sup>29</sup> Cette dernière est volatile (Ou non persistante).



 Nombre de pages : 61	<h1>Concevoir un site web avec PHP et MySQL</h1>	Réalisé le :	28-12-2024
		Modifié le :	28/12/2024

On peut modifier quand on veut la valeur de cette variable, faire des opérations dessus, etc. Et quand on en a besoin, on l'appelle (par son nom), et elle nous dit gentiment la valeur qu'elle contient.

### 2.1.1.2. Découvrir les différents types de variables

Les variables sont capables de stocker différents types d'informations. On parle de **types de données**. Voici les principaux types à connaître.

- **Les chaînes de caractères (string)** : c'est le nom informatique qu'on donne au texte.
- **Les nombres entiers (int)** : ce sont les nombres du type 1, 2, 3, 4, etc. 🖱️ **On compte aussi parmi eux les entiers relatifs** : -1, -2, -3...
- **Les nombres décimaux (float)** : ce sont les nombres à virgule, comme 14,738. Attention, 🖱️ **les nombres doivent être écrits avec un point** au lieu de la virgule (c'est la notation anglaise).
- **Les booléens (bool)** : c'est un type très important qui permet de stocker soit vrai soit faux.
- **Rien (NULL)** : aussi bizarre que cela puisse paraître, on a parfois besoin de dire qu'une variable ne contient rien. Ce n'est pas vraiment un type de données, mais plutôt **l'absence** de type.

Valeur	Type	Exemple
Chaîne de caractères	string	\$authorName = 'Mathieu';
Nombres entiers	int	\$authorAge = 3;
Nombres décimaux	float	\$productPrice = 14.738;
Booléens	bool	\$isAllowed = true;
Rien	NULL	\$authorQuality = NULL;

Figure 11: Les variables

#### 2.1.1.2.1. PHP est un langage de programmation à typage dynamique faible



<sup>30</sup>Le typage dynamique faible est un concept clé en programmation qui concerne la manière dont un langage de programmation gère les types de données.

Décomposons ce concept en deux parties :

- **Typage dynamique**

Un langage de programmation est dit **dynamique** lorsqu'il détermine les types des variables à **l'exécution** plutôt qu'à la compilation. Cela signifie que :

<sup>30</sup> <https://chatgpt.com/share/67630ee2-dac8-8000-9805-90bae5528380>

	<h1>Concevoir un site web avec PHP et MySQL</h1>	Réalisé le :	28-12-2024
Nombre de pages : 61		Modifié le :	28/12/2024

- Vous n'avez pas besoin de déclarer explicitement le type d'une variable lorsque vous la créez.
- Une variable peut changer de type au cours de l'exécution du programme.

👉 En PHP, le type des variables est déterminé à l'**exécution**, et une variable **peut changer de type** en cours d'exécution sans lever d'erreur.

- **Typage faible**

Un langage est dit **faiblement typé** lorsque les **conversions implicites** entre différents types de données sont permises. Ces conversions peuvent se produire automatiquement sans intervention explicite du programmeur. Cela peut entraîner des comportements inattendus si les conversions ne sont pas bien comprises.

👉 PHP permet des conversions implicites entre différents types selon le contexte. Ce qui peut simplifier certaines tâches mais aussi introduire des comportements imprévisibles.



#### Avantages :

Simplicité et flexibilité.

Moins de code requis pour gérer explicitement les types.

#### Inconvénients :

Les comportements inattendus dus aux conversions automatiques.

Les erreurs peuvent être plus difficiles à diagnostiquer.

### 2.1.2. Affecter une valeur à une variable

Regardez ce code d'exemple :

```
<?php
$userAge = 17;
?>
```

Avec ce code PHP, on vient en fait de créer une variable :

son nom est `userAge` ;

sa valeur est `17` .


👉 Notez qu'on ne peut pas mettre d'espace dans un nom de variable. On utilise donc une majuscule pour "détacher" visuellement les mots et les rendre plus lisibles.

C'est ce que l'on appelle la convention camelCase (cela fait référence aux bosses d'un chameau).



**PHP est case sensitive** . i.e., il tient compte des majuscules et minuscules.

	<h1>Concevoir un site web avec PHP et MySQL</h1>	Réalisé le :	28-12-2024
Nombre de pages : 61		Modifié le :	28/12/2024

 Quand vous nommez des variables, évitez les accents, les cédilles et tout autre symbole : PHP ne les apprécie pas trop...  
C'est pour cela que j'ai écrit `age` et non `âge` .


Analysons dans le détail le code qu'on vient de voir.

D'abord, on écrit le symbole "dollar" (\$) : il précède toujours le nom d'une variable. C'est comme un signe de reconnaissance, si vous préférez : ça permet de dire à PHP "J'utilise une variable".

Ensuite, il y a le signe "égal" (=) : celui-là, c'est logique, c'est pour dire que `$userAge` est égal à...

À la suite, il y a la valeur de la variable, ici `17` .

Enfin, il y a l'incontournable point-virgule (;) qui permet de terminer l'instruction.

 Rien ne s'affiche tant que vous n'utilisez pas `echo` .  
Là, le serveur a juste créé la variable temporairement en mémoire, mais il n'a rien fait d'autre.

Supposons maintenant que l'on écrive ceci :

```
<?php
$userAge = 17; // La variable est créée et vaut 17
$userAge = 23; // La variable est modifiée et vaut 23
$userAge = 55; // La variable est modifiée et vaut 55
?>
```

La variable `$userAge` va être créée et prendre pour valeur, dans l'ordre : 17, 23, puis 55.


Tout cela va très vite : l'ordinateur étant très rapide, vous n'aurez pas le temps de dire « ouf » que tout ce code PHP aura été exécuté.

Comme tout à l'heure, rien ne s'affiche. Seulement, quelque part dans la mémoire de l'ordinateur, une petite zone nommée `userAge` vient de prendre la valeur 17, puis 23, puis 55.

### 2.1.3. Utiliser les types de données

Voici un exemple de variable pour chacun des types.

#### 2.1.3.1. Le type *string* (chaîne de caractères)

 Ce type permet de stocker du texte.

Pour cela, vous devez entourer votre texte de :

guillemets doubles<sup>31</sup> `" "` ;

---

<sup>31</sup> ASCII = 34

	<h1>Concevoir un site web avec PHP et MySQL</h1>	Réalisé le :	28-12-2024
Nombre de pages : 61		Modifié le :	28/12/2024

ou de guillemets simples<sup>32</sup> '' (attention, ce sont des apostrophes).

Par exemple, pour conserver en mémoire les noms et mails de nos utilisateurs :



```
<?php
$fullname = "Mathieu Nebra";
$email = 'mathieu.nebra@exemple.com';
?>
```



Attention, petit piège : si vous voulez insérer un guillemet simple alors que le texte est entouré de guillemets simples, il faut l'**échapper** en insérant un **antislash** devant. Il en va de même pour les guillemets doubles.

Voici un exemple pour bien comprendre :


```
<?php
$variable = "Mon \"nom\" est Mathieu";
$variable = 'Je m\'appelle Mathieu';
?>
```

En effet, si vous oubliez de mettre un antislash, PHP va croire que c'est la fin de la chaîne et il ne  comprendra pas le texte qui suivra (et vous aurez en fait un message  Parse error).

Vous pouvez en revanche insérer sans problème des guillemets simples au milieu de guillemets doubles, et inversement :

```
<?php
$variable = 'Mon "nom" est Mathieu';
$variable = "Je m'appelle Mathieu";
?>
```



La différence est subtile, faites attention. Il y a d'ailleurs une différence plus importante entre les deux types de guillemets, dont  **nous parlerons plus loin**.

#### 2.1.3.2. Le type *int* (nombre entier)

Il suffit d'écrire le nombre que vous voulez stocker, sans guillemets :

```
<?php
$userAge = 17;
?>
```

#### 2.1.3.3. Le type *float* (nombre décimal)

Vous devez écrire votre nombre avec un point au lieu d'une virgule. C'est la notation anglaise.

```
<?php
$price = 57.3;
?>
```

<sup>32</sup> ASCII = 39 à ne pas confondre avec le backtick ` (accent grave) qui le code ASCII = 96.

	<h1>Concevoir un site web avec PHP et MySQL</h1>	Réalisé le :	28-12-2024
Nombre de pages : 61		Modifié le :	28/12/2024

#### 2.1.3.4. Le type `bool` (booléen)

Pour dire si une variable vaut vrai ou faux, vous devez écrire le mot `true` ou `false` sans guillemets autour (ce n'est pas une chaîne de caractères !) :

```
<?php
$isAuthor = true;
$isAdministrator = false;
?>
```

#### 2.1.3.5. Une variable vide avec `NULL`

Si vous voulez créer une variable qui ne contient rien, vous devez lui passer le mot-clé `NULL` (vous pouvez aussi l'écrire en minuscules : `null` ).

Cela sert simplement à indiquer que la variable ne contient rien, tout du moins 📁 pour le moment.

#### 2.1.4. Afficher le contenu d'une variable

Vous vous souvenez que l'on peut afficher du texte avec `echo` ?

On peut aussi s'en servir pour afficher la valeur d'une variable !

```
<?php
$fullname = 'Mathieu Nebra';
echo $fullname;
?>
```

Quand il s'agit d'une variable, on ne met pas de guillemets autour, pour l'afficher.

#### 2.1.5. Concaténer une variable

En fait, écrire "Mathieu Nebra" tout seul comme on l'a fait n'est pas très parlant. On aimerait écrire du texte autour pour dire : « Bienvenue Mathieu Nebra ».

La **concaténation** est justement **un moyen d'assembler du texte et des variables**.

Pour cela, il y a deux méthodes :

1. Avec des guillemets simples.
2. Ou avec guillemets doubles.

Et c'est là qu'on va voir, **qu'entre les deux, il y a 📁 une différence !**

##### 2.1.5.1. Effectuez l'interpolation avec des guillemets doubles

📁 **L'interpolation** en PHP<sup>33</sup> vous permet d'inclure directement des variables dans une chaîne de caractères sans avoir à les concaténer séparément, ce qui rend votre code plus lisible et concis.

Concrètement, essayez ce code :

<sup>33</sup> L' Interpolation des chaînes de caractères existe aussi dans d'autres langages, comme Java ou JavaScript, par exemple.

	<h1>Concevoir un site web avec PHP et MySQL</h1>	Réalisé le :	28-12-2024
Nombre de pages : 61		Modifié le :	28/12/2024

```
<?php
    $fullname = "Mathieu Nebra";
    echo "Bonjour {$fullname} et bienvenue sur le site !";
?>
```



Attention à ne pas oublier d'entourer la variable d'accolades 🖱️ pour éviter les erreurs.

Dans cet exemple, `echo "Bonjour, $fullname123!";` nous aurons un message d'erreur indiquant que la variable `$fullname123` n'est pas définie.

Maintenant, si nous ajoutons les accolades `echo "Bonjour, {$fullname}123!";` alors la variable est interprétée correctement et nous aurons `"Bonjour, Mathieu Nebra123!"`.

En effet, lorsque vous utilisez des **guillemets doubles**, les variables qui se trouvent à l'intérieur sont **analysées et remplacées** par leur vraie valeur.

#### 2.1.5.2. Concaténez avec des guillemets simples

Si vous écrivez le code précédent entre guillemets simples, vous allez avoir une drôle de surprise :

```
<?php
    $fullname = 'Mathieu Nebra';
    echo 'Bonjour {$fullname} et bienvenue sur le site !'; // ERREUR !
?>
```

Ça affiche 🔥 : `Bonjour {$fullname}` et bienvenue sur le site ! .

🖱️ Pour éviter de lever une erreur, il va falloir :

- écrire la variable en dehors des guillemets et
- séparer les éléments les uns des autres à l'aide d'un point.

Regardez :

```
<?php
    $fullname = 'Mathieu Nebra';
    echo 'Bonjour ' . $fullname . ' et bienvenue sur le site !'; // OK
?>
```

Cette fois, ça affiche bien comme on voulait.

En règle générale, on utilise l'interpolation lorsqu'on a besoin **d'incorporer des variables directement dans une chaîne de caractères** de manière propre et concise. C'est idéal pour rendre le code plus lisible.

En revanche, s'il y a des **expressions complexes, des conditions ou des opérations à effectuer pendant la concaténation**, alors la concaténation traditionnelle avec des guillemets simples reste une option solide.

L'essentiel est de choisir l'approche qui rend le code le plus clair et le plus maintenable en fonction du contexte du projet.

	<h1>Concevoir un site web avec PHP et MySQL</h1>	Réalisé le :	28-12-2024
Nombre de pages : 61		Modifié le :	28/12/2024

### 2.1.5.3. To do : les opérateurs de chaînes de caractères

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

### 2.1.6. Faire des calculs simples

Ici, comme vous vous en doutez, on ne va travailler que sur des variables qui contiennent des nombres.

#### 2.1.6.1. Les opérations de base : addition, soustraction...

Les signes à connaître pour faire les quatre opérations de base sont représentés par le tableau suivant.

Symbole	Signification
+	Addition
-	Soustraction
*	Multiplication
/	Division
%	Modulo

En complément vous avez l'opération modulo, c'est-à-dire le reste d'une division euclidienne.

Après, pour vous en servir, ça coule de source. Voici quelques exemples :

```
<?php
$number = 2 + 4; // $number prend la valeur 6
$number = 5 - 1; // $number prend la valeur 4
$number = 3 * 5; // $number prend la valeur 15
$number = 10 / 2; // $number prend la valeur 5

// Allez on rajoute un peu de difficulté
$number = 3 * 5 + 1; // $number prend la valeur 16
$number = (1 + 2) * 2; // $number prend la valeur 6
?>
```

Seulement, il ne faut pas craindre de « jongler » avec les variables.

Voici des calculs avec plusieurs variables :

```
<?php
$number = 10;
$result = ($number + 5) * $number; // $result prend la valeur 150
?>
```

#### 2.1.6.2. Le modulo

Il est possible de faire un autre type d'opération un peu moins connu : le modulo<sup>34</sup>.

Cela représente le reste de la division entière.

<sup>34</sup> Utile pour certains algorithmes

	<h1>Concevoir un site web avec PHP et MySQL</h1>	Réalisé le :	28-12-2024
Nombre de pages : 61		Modifié le :	28/12/2024

Le modulo permet justement de récupérer ce reste :

```
<?php
$number = 10 % 5; // $number prend la valeur 0 car la division tombe juste
$number = 10 % 3; // $number prend la valeur 1 car il reste 1
?>
```

Je passe sous silence les opérations plus complexes telles que :

la racine carrée ;  
l'exponentielle ;  
la factorielle ;  
etc.

Toutes ces opérations peuvent être réalisées en PHP mais il faudra passer par ce qu'on appelle des **fonctions**, une notion que l'on découvrira plus tard.

Les opérations basiques que l'on vient de voir sont amplement suffisantes pour la programmation PHP de tous les jours.

#### 2.1.7. En résumé

- Une variable est une petite information qui reste stockée en mémoire le temps de la génération de la page PHP. Elle a un nom et une valeur.
- Il existe plusieurs types de variables qui permettent de stocker différents types d'informations : du texte ( `string` ), des nombres entiers ( `int` ), des nombres décimaux ( `float` ), des booléens pour stocker vrai ou faux ( `bool` ), etc.
- En PHP, un nom de variable commence par le symbole dollar : `$age` , par exemple.
- La valeur d'une variable peut être affichée avec l'instruction `echo` .
- Il est possible de faire des calculs mathématiques entre plusieurs variables : addition, soustraction, multiplication...

## 2.2. Adapter le comportement de l'application à l'aide des conditions

Ce chapitre est fondamental. En effet, vous serez très souvent amené à employer des conditions dans vos pages web PHP.

Dans notre projet fil rouge de création de site de recettes, on voudra afficher des informations en fonction du contexte. Par exemple :

- autoriser l'auteur d'une recette à la modifier, mais pas les autres utilisateurs ;
- afficher seulement la liste des recettes qui auront été vérifiées par un administrateur ;
- etc.

👉 Les conditions, c'est ce qui va réellement vous permettre de créer une application dynamique.



 Nombre de pages : 61	<h1>Concevoir un site web avec PHP et MySQL</h1>	Réalisé le :	28-12-2024
		Modifié le :	28/12/2024

À la fin de ce chapitre, vous aurez les bases nécessaires pour autoriser l'affichage d'une recette ou l'accès d'un utilisateur à une page 📁 **selon les conditions de votre application**<sup>35</sup>.

### 2.2.1. Appropriiez-vous la structure de base : if... else

Une condition peut être écrite en PHP sous différentes formes.



On parle de **structures conditionnelles**

#### 2.2.1.1. Retenir les symboles à connaître

Voici les symboles que nous serons amenés à utiliser. Essayez de bien les retenir, ils vous seront utiles :

Symbole	Signification
===	Est égal à
>	Est supérieur à
<	Est inférieur à
>=	Est supérieur ou égal à
<=	Est inférieur ou égal à
!=	Est différent de



Il y a trois symboles « égal » ( === ) sur la première ligne.

Il ne faut pas confondre ça avec le simple = (que je vous ai appris dans le chapitre sur les variables). Ici, le triple égal sert à tester l'égalité, à dire « Si c'est égal à... ». Dans les conditions, on utilisera toujours le triple égal ( === )<sup>36</sup>.

#### 2.2.1.1.1. To do : les opérateurs PHP en carte heuristique (Xmind)

(et en html OR Xmind/html ?)

#### 2.2.1.1.2. To do : Précédence des opérateurs en PHP

#### 2.2.1.2. Utiliser la structure if... else

Voici ce qu'on doit écrire, dans l'ordre, pour utiliser cette condition.

- Pour introduire une condition, on utilise le mot if qui signifie « si », en anglais.


<sup>35</sup> Application métier, donc comporte cette dernière comporte des règles de gestion au sens de la méthode Merise.

<sup>36</sup> Plus précisément : permet de tester l'égalité en termes de valeurs et de types

	<h1>Concevoir un site web avec PHP et MySQL</h1>	Réalisé le :	28-12-2024
Nombre de pages : 61		Modifié le :	28/12/2024

- On ajoute à la suite entre parenthèses la condition en elle-même (vous allez voir que vous pouvez inventer une infinité<sup>37</sup> ( $\infty$ )<sup>38</sup> de conditions<sup>39</sup>).
- Enfin, on ouvre des accolades à l'intérieur desquelles on placera les instructions à exécuter si la condition est remplie.

Puisqu'un exemple vaut toujours mieux qu'un long discours :



```
<?php
$isEnabled = true; // The access condition
if ($isEnabled === true) {
    echo "You are authorized to access the site ";
}
?>
```

Ce qui compte ici, c'est qu'il y a deux possibilités<sup>40</sup> :

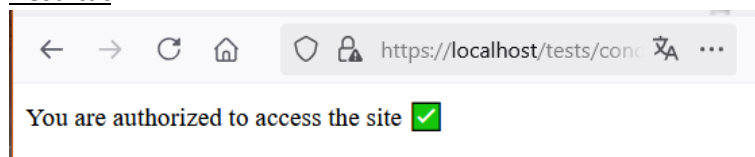
1. Soit la condition est remplie et alors on affiche quelque chose.
2. Sinon, on saute les instructions entre accolades, on ne fait rien.

Bon, on peut quand même améliorer notre exemple :

```
<?php
$isEnabled = true; //La condition d'accès
// $isEnabled = false;

if ($isEnabled === true) {
    echo "You are authorized to access the site ";
}
else{
    echo "Access denied ";
}
?>
```

**Résultat :**



Comment fonctionne ce code ?

Tout d'abord, j'ai mis plusieurs instructions entre accolades. Ensuite, vous avez remarqué que j'ai ajouté le mot `else` (« sinon »).

Essayez ce bout de code en modifiant la valeur de `$isEnabled` (sur la première ligne).

<sup>37</sup> Quel ordre de grandeur ?


<sup>38</sup> Alt+8734

<sup>39</sup>  $\infty$  en potentiel ou en acte ?

<sup>40</sup> Donc une alternative


	<h1>Concevoir un site web avec PHP et MySQL</h1>	Réalisé le :	28-12-2024
Nombre de pages : 61		Modifié le :	28/12/2024

```
<?php
# $isEnabled = true; // La condition d'accès
$isEnabled = false;

if ($isEnabled === true) {
    echo "You are authorized to access the site 

```



Access denied 

👉 Ici, par exemple, j'ai donné une valeur différente à la variable `$isAllowedToEnter` après avoir affiché un message (une valeur qui pourrait nous servir par la suite) :

```
<?php
$isAllowedToEnter = "Oui";


// SI on a l'autorisation d'entrer
if ($isAllowedToEnter === "Oui") {
    // instructions à exécuter quand on est autorisé à entrer
} // SINON SI on n'a pas l'autorisation d'entrer
elseif ($isAllowedToEnter === "Non") {
    // instructions à exécuter quand on n'est pas autorisé à entrer
} // SINON (la variable ne contient ni Oui ni Non, on ne peut pas agir)
else {
    echo "Euh, je ne comprends pas ton choix, tu peux me le rappeler s'il te plaît ?";
}
?>
```

La principale nouveauté ici, c'est le mot-clé `elseif` qui signifie « sinon si ».

Dans l'ordre, PHP rencontre les conditions suivantes :

1. Si `$isAllowedToEnter` est égale à « Oui », tu exécutes ces instructions...
2. Sinon, si `$isAllowedToEnter` est égale à « Non », tu exécutes ces autres instructions...
3. Sinon, tu redemandes l'âge pour savoir si on a ou non l'autorisation d'entrer.

	<h1>Concevoir un site web avec PHP et MySQL</h1>	Réalisé le :	28-12-2024
Nombre de pages : 61		Modifié le :	28/12/2024

 Pour vérifier si la variable est vide, vous pouvez taper : `if ($variable === NULL)` <sup>41</sup>

### 2.2.1.3. Étudier le cas des booléens

Si on regarde bien le dernier code source (avec `$isAllowedToEnter`), il serait plus adapté d'utiliser des booléens.

Les booléens sont ces variables qui valent :

- soit `true` (vrai) ;
- soit `false` (faux).

Voici comment on teste une variable booléenne :

```
<?php
$isAllowedToEnter = true;


if ($isAllowedToEnter) {
    echo "Bienvenue petit nouveau. :o)";
}
else {
    echo "T'as pas le droit d'entrer !";
}
?>
```

L'un des avantages des booléens, c'est qu'ils sont particulièrement adaptés aux conditions.

Pourquoi ?

Parce qu'en fait vous n'êtes pas obligé d'ajouter le `=== true`.

PHP comprend qu'il faut qu'il vérifie si `$isAllowedToEnter` **vaut** `true`.

 Les avantages des booléens :

- c'est plus rapide à écrire pour vous ;
- ça se comprend bien mieux.

👉 C'est un **raccourci à connaître** quand on travaille sur des booléens.

Il y a un symbole qui permet de vérifier si la variable vaut `false` : le point d'exclamation (!).

On écrit :

```
<?php
$isAllowedToEnter = true;

// Si pas autorisé
if (! $isAllowedToEnter) {
}
?>
```

<sup>41</sup> Il faut faire cette vérification en début de code ! 

	<h1>Concevoir un site web avec PHP et MySQL</h1>	Réalisé le :	28-12-2024
Nombre de pages : 61		Modifié le :	28/12/2024



C'est une autre façon de faire.

Si vous préférez mettre `if ($isAllowedToEnter === false)` c'est tout aussi bien<sup>42</sup>, mais la méthode « courte » est plus lisible.

#### 2.2.1.4. Poser des conditions multiples

Ce qu'on va essayer de faire, c'est de poser plusieurs conditions à la fois. Pour cela, on aura besoin de nouveaux mots-clés. Voici les principaux à connaître :


Mot-clé	Signification	Symbole équivalent
AND	Et	&&
OR	Ou	



Le symbole équivalent pour OR est constitué de deux barres verticales. Cette barre verticale<sup>43</sup>, ou « pipe<sup>44</sup> », en anglais.

##### 2.2.1.4.1. Premier exemple avec &&

```
<?php
$isEnabled = true;
$isOwner = false;

if ($isEnabled && $isOwner) {
    echo 'Accès à la recette validé 


```

C'est tout simple, en fait, et ça se comprend très bien : si l'utilisateur est actif et qu'il est l'auteur, il peut accéder à la recette validée. Sinon, il verra s'afficher un message de refus.

##### 2.2.1.4.2. Deuxième exemple avec ||

Un exemple avec || pour que vous l'ayez vu au moins une fois.

```
<?php
$isEnabled = true;
$isOwner = false;
$isAdmin = true;

if (($isEnabled && $isOwner) || $isAdmin) {
    echo 'Accès à la recette validé 

```

<sup>42</sup> Mais cela rend le code source plus verbeux.

<sup>43</sup> Pour taper une barre verticale, appuyer sur les touches « Alt Gr » et « 6 » d'un clavier AZERTY.

<sup>44</sup> Code ascii du pipe : 124.

	<h1>Concevoir un site web avec PHP et MySQL</h1>	Réalisé le :	28-12-2024
Nombre de pages : 61		Modifié le :	28/12/2024

Nous rajoutons une condition supplémentaire : soit la condition précédente s'applique, soit l'utilisateur concerné est un administrateur.

#### 2.2.1.5. Utiliser cette astuce bonus

Avec les conditions, il y a une astuce à connaître.

Sachez que les deux codes ci-dessous donnent exactement le même résultat :

##### ► code 1 :

```
<?php
$chickenRecipesEnabled = true;
if ($chickenRecipesEnabled) {
    echo '<h1>Liste des recettes à base de poulet</h1>';
}
?>
```

##### ► code 2 :

```
<?php $chickenRecipesEnabled = true; ?>
<?php if ($chickenRecipesEnabled): ?> <!-- Ne pas oublier le ":" -->
<h1>Liste des recettes à base de poulet</h1>
<?php endif; ?><!-- Ni le ";" après le endif -->
```

Dans le second cas on n'a pas utilisé de `echo` .

La syntaxe pour utiliser la condition diffère un peu :

- Il n'y a pas d'accolade.
- On ajoute : après la parenthèse fermante de l'instruction `if` .
- Et il faut ajouter une instruction `endif` ; .

#### 2.2.2. Utilisez la condition switch pour optimiser votre code

En théorie, les structures à base de `if... elseif... else` que je viens de vous montrer suffisent pour traiter n'importe quelle condition.

Une autre structure :

Pour vous montrer l'intérêt de `switch` ! Vous allez bientôt comprendre...

Regardez cet exemple à base de `if` et de `elseif` :

```
<?php
$grade = 16;
if ($grade === 0) {
    echo "Il faudra revoir tout le cours !";
}

elseif ($grade === 5) {
    echo "Tu dois réviser plusieurs chapitres";
}

elseif ($grade === 7) {
    echo "Il te manque quelques révisions pour atteindre la moyenne";
}
```

	<h1>Concevoir un site web avec PHP et MySQL</h1>	Réalisé le :	28-12-2024
Nombre de pages : 61		Modifié le :	28/12/2024

```

}
elseif ($grade === 10) {
    echo "Tu as pile poil la moyenne, c'est un peu juste...";
}
elseif ($grade === 12) {
    echo "Tu es assez bon";
}
elseif ($grade === 16) {
    echo "Tu te débrouilles très bien !";
}
elseif ($grade === 20) {
    echo "Excellent travail, c'est parfait !";
}
else {
    echo "Désolé, je n'ai pas de message à afficher pour cette note";
}
?>

```

Comme on peut le constater : c'est lourd, long, et 🙄répétitif.<sup>45</sup>

Dans ce cas, on peut utiliser une autre structure plus souple : c'est `switch` !

Voici le même exemple avec `switch` (le résultat est le même, mais le code est plus adapté) :

```

<?php
$grade = 10;

switch ($grade) // on indique sur quelle variable on travaille
{
    case 0: // dans le cas où $grade vaut 0
        echo "Il faudra revoir tout le cours !";
        break;

    case 5: // dans le cas où $grade vaut 5
        echo "Tu dois réviser plusieurs modules";
        break;

    case 7: // dans le cas où $grade vaut 7
        echo "Il te manque quelques révisions pour atteindre la moyenne ";
        break;

    case 10: // etc. etc.
        echo "Tu as pile poil la moyenne, c'est un peu juste...";
        break;

    case 12:
        echo "Tu es assez bon";
        break;
}

```

<sup>45</sup> Cela va à l'encontre du principe « DRY » : "Don't Repeat Yourself" ; Un des principes de programmation.

	<h1>Concevoir un site web avec PHP et MySQL</h1>	Réalisé le :	28-12-2024
Nombre de pages : 61		Modifié le :	28/12/2024



```

case 16:
    echo "Tu te débrouilles très bien !";
    break;

case 20:
    echo "Excellent travail, c'est parfait !";
    break;

default:
    echo "Désolé, je n'ai pas de message à afficher pour cette note";
}
?>

```

 Tester ce code : changer la note (dans la première instruction) pour voir comment PHP réagit ! Et si vous voulez apporter quelques modifications à ce code (vous allez voir qu'il n'est pas parfait), n'hésitez pas, ça vous fera de  l'entraînement !<sup>46</sup>

Bon alors, qu'est-ce qui est différent ?



- Il y a beaucoup moins d'accolades : elles marquent seulement le début et la fin du `switch` .
- On indique au début du `switch` sur quelle variable on travaille, ici `$grade`. On dit à PHP :


Je vais analyser la valeur de `$grade` .


- On utilise des `case` pour analyser chaque cas : `case 0` , `case 10` , etc. Cela signifie :


Dans le cas où la valeur est 0... Dans le cas où la valeur est 10...


Quel est l'avantage d'utiliser `switch` ?

 On n'a plus besoin de mettre le triple égal ! 

 Le `switch` ne peut tester que l'égalité.

En effet : cela ne marche pas  avec les autres symboles : `<` `>` `<=` `>=` `!==`

 Le mot-clé `default` à la fin est un peu l'équivalent du `else` .

 Il y a cependant une chose importante à savoir :



Supposons dans notre exemple que la **note soit de 10**.


PHP va lire :

`case 0` ? Non. Je saute.

`case 5` ? Non plus. Je saute.


`case 7` ? Non plus. Je saute.

`case 10` ? Oui, j'exécute les instructions.


<sup>46</sup> Car c'est en codant qu'on apprend à coder 



	<h1>Concevoir un site web avec PHP et MySQL</h1>	Réalisé le :	28-12-2024
Nombre de pages : 61		Modifié le :	28/12/2024

 Mais contrairement aux `elseif`, PHP ne s'arrête pas là et continue à lire les instructions des cases qui suivent !

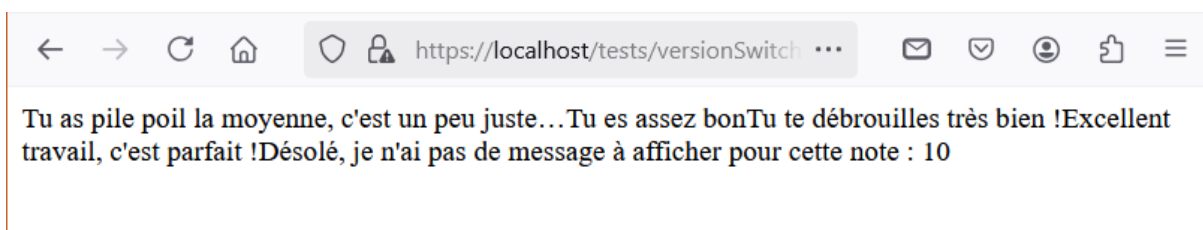
- `case 12`
- `case 16`
- `etc.`

 Pour empêcher cela, utilisez l'instruction `break` !


👉 En effet, l'instruction `break` demande à PHP de sortir du `switch`.

Dès que PHP tombe sur `break`, il sort des accolades et donc il ne lit pas les case qui suivent.

Essayez d'enlever les `break` dans le code précédent, vous allez comprendre pourquoi ils sont indispensables !



En effet ! 😊


 Quand doit-on choisir `if` ?  
Et quand doit-on choisir `switch` ?

C'est surtout un problème de présentation et de clarté :

1. Pour une condition simple et courte, on utilise le `if`.
2. Et quand on a une série de conditions à analyser, on préfère utiliser `switch` pour rendre le code plus clair.

### 2.2.3. Découvrez les ternaires : des conditions condensées

Il existe une autre forme de condition. Il s'agit de ce qu'on appelle les **ternaires**.

 Un **ternaire** est une condition condensée qui sert à faire **deux choses sur une seule ligne** :

1. Tester la valeur d'une variable dans une condition.
2. Affecter une valeur à une variable selon que la condition est vraie ou non<sup>47</sup>.

Prenons cet exemple à base de `if... else` qui met un booléen `$majeur` à vrai ou faux selon l'âge du visiteur :

<sup>47</sup> True ou False dans le code.

	<h1>Concevoir un site web avec PHP et MySQL</h1>	Réalisé le :	28-12-2024
Nombre de pages : 61		Modifié le :	28/12/2024

```
<?php
$userAge = 24;

if ($userAge >= 18) {
    $isAdult = true;
}
else {
    $isAdult = false;
}
?>
```

On peut faire la même chose en une seule ligne grâce à une structure ternaire :

```
<?php
$userAge = 24;

$isAdult = ($userAge >= 18) ? true : false;

// Ou mieux, dans ce cas précis
$isAdult = ($userAge >= 18);
?>
```

Ici, tout notre test précédent a été fait sur une seule ligne !

La condition testée est `$userAge >= 18` .

Si c'est vrai, alors la valeur indiquée après le point d'interrogation (ici `true`) sera affectée à la variable `$isAdult` .

Sinon, c'est la valeur qui suit le symbole : (ici `false` ) qui sera affectée à `$isAdult` .

C'est un peu tordu, mais ça marche.



Il faut avouer que les ternaires sont un peu difficiles à lire car ils sont très condensés. Mais sachez les reconnaître et les comprendre, 🖱️ **si vous en rencontrez un jour**<sup>48</sup> en lisant le code source de quelqu'un d'autre.

#### 2.2.4. En résumé

- Les conditions permettent à PHP de prendre des décisions en fonction de la valeur des variables.
- La forme de condition la plus courante est `if ... elseif ... else` qui signifie « si »... « sinon si »... « sinon ».
- On peut combiner des conditions avec les instructions `&&` (« et ») et `||` (« ou »).
- Si une condition comporte de nombreux `elseif` , il peut être plus pratique d'utiliser la condition `switch`.
- **Les ternaires sont des conditions condensées** qui font un test sur une variable, et en fonction des résultats de ce test, elles donnent une valeur à une autre variable. Elles sont cependant plus rarement utilisées.

<sup>48</sup> C'est souvent le cas 😊, car le précédent développeur a encore été remplacé 😞

	<h1>Concevoir un site web avec PHP et MySQL</h1>	Réalisé le :	28-12-2024
Nombre de pages : 61		Modifié le :	28/12/2024

2.3. Afficher une liste de recettes à l'aide des boucles

2.4. Organiser les données à l'aide des tableaux

2.5. Exploiter toute la puissance des fonctions PHP !

2.6. Au secours ! Mon script plante !

2.7. Organiser les pages du site en blocs fonctionnels

2.8. Quiz : Réaliser un site web dynamique avec PHP

## 3. Partie 3 - Transmettre des données de page en page

3.1. Écouter la requête des utilisateurs grâce aux URL

3.2. Administrer des formulaires de façon sécurisée

3.3. Activer le partage de fichiers

3.4. Implémenter un système de connexion

3.5. Conserver des données grâce aux sessions et aux cookies

3.6. Quiz : Transmettre des données de page en page

## 4. Partie 4 - Stockez des informations dans une base de données

	<h1>Concevoir un site web avec PHP et MySQL</h1>	Réalisé le :	28-12-2024
Nombre de pages : 61		Modifié le :	28/12/2024

4.1. Travailler avec une base de données

4.2. Mettre en place une base de données avec phpMyAdmin

4.3. Accéder aux données en PHP avec PDO

4.4. Ajouter, modifier et supprimer des recettes !

4.5. Ajouter des commentaires grâce aux jointures SQL

4.6. Aller plus loin

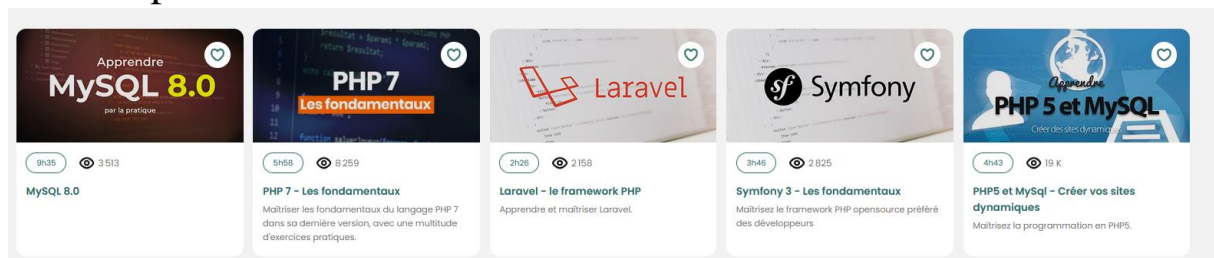
4.7. Quiz : Utiliser une base de données

4.8. Le Certificat de Réussite

## 5. Annexes

5.1. Table ascii sur 7 bits

## 6. Aller plus loin



### 6.1. Cours

Mettez en ligne votre site web : 2 🕒 1 partie

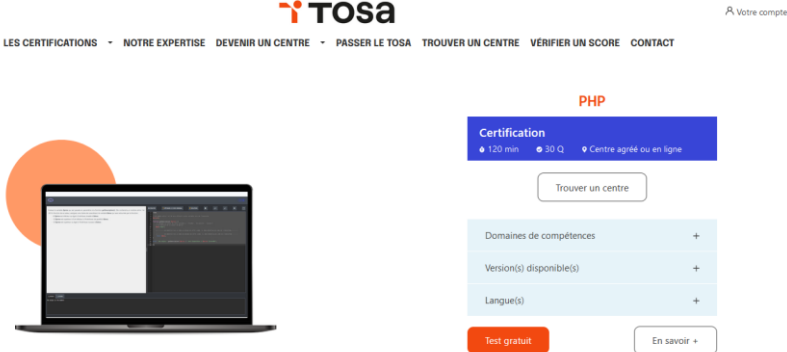
<https://openclassrooms.com/fr/courses/7192596-mettez-en-ligne-votre-site-web>

Initiez-vous à Linux : 8 🕒 4 parties

	<h1>Concevoir un site web avec PHP et MySQL</h1>	Réalisé le :	28-12-2024
Nombre de pages : 61		Modifié le :	28/12/2024

<https://openclassrooms.com/fr/courses/7170491-initiez-vous-a-linux>

## 6.2. Certifications TOSA



The screenshot shows the TOSA website interface. At the top, there's a navigation bar with links: LES CERTIFICATIONS, NOTRE EXPERTISE, DEVENIR UN CENTRE, PASSER LE TOSA, TROUVER UN CENTRE, VERIFIER UN SCORE, and CONTACT. Below this, there's a section for PHP certification. It features a blue header with 'Certification' and '120 min 30 Q Centre agréé ou en ligne'. A 'Trouver un centre' button is present. Below that, there are expandable sections for 'Domaines de compétences', 'Version(s) disponible(s)', and 'Langue(s)'. At the bottom, there are two buttons: 'Test gratuit' and 'En savoir +'. To the left of the certification details, there's an image of a laptop displaying code.

<https://www.tosa.org/FR/certification?brand=code>

## 7. End