

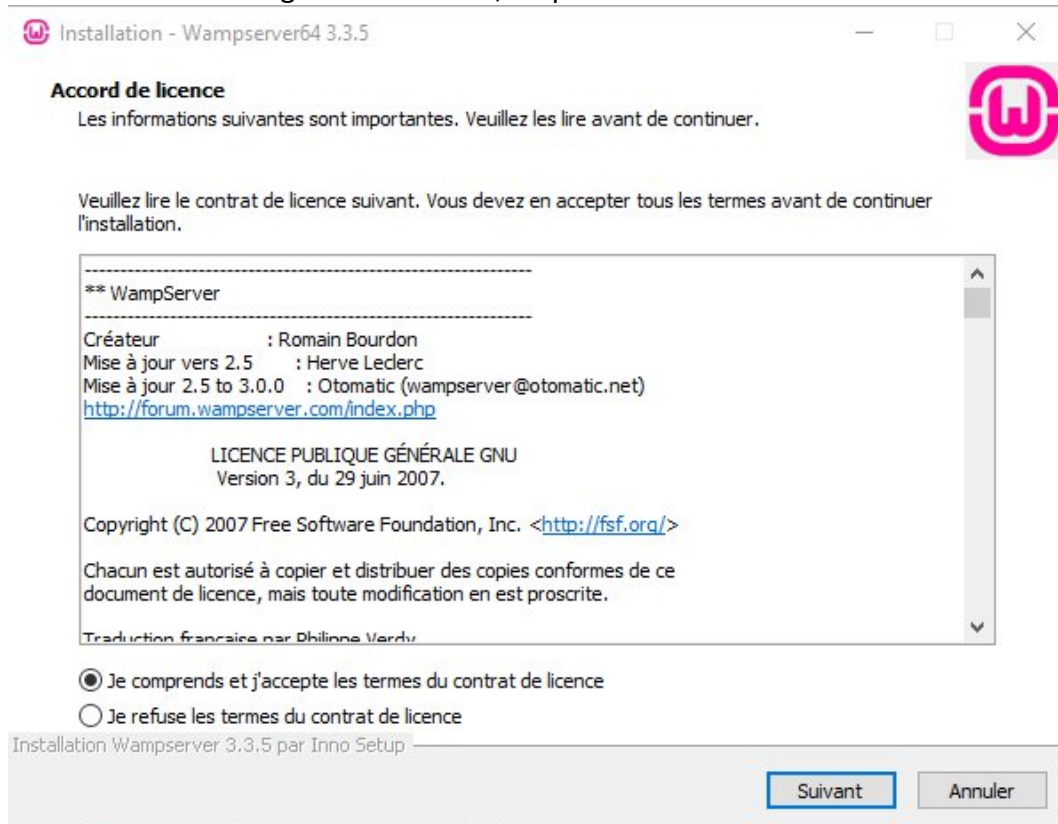
TDs

TD 1 : installation d'un serveur local, utilisation d'un IDE

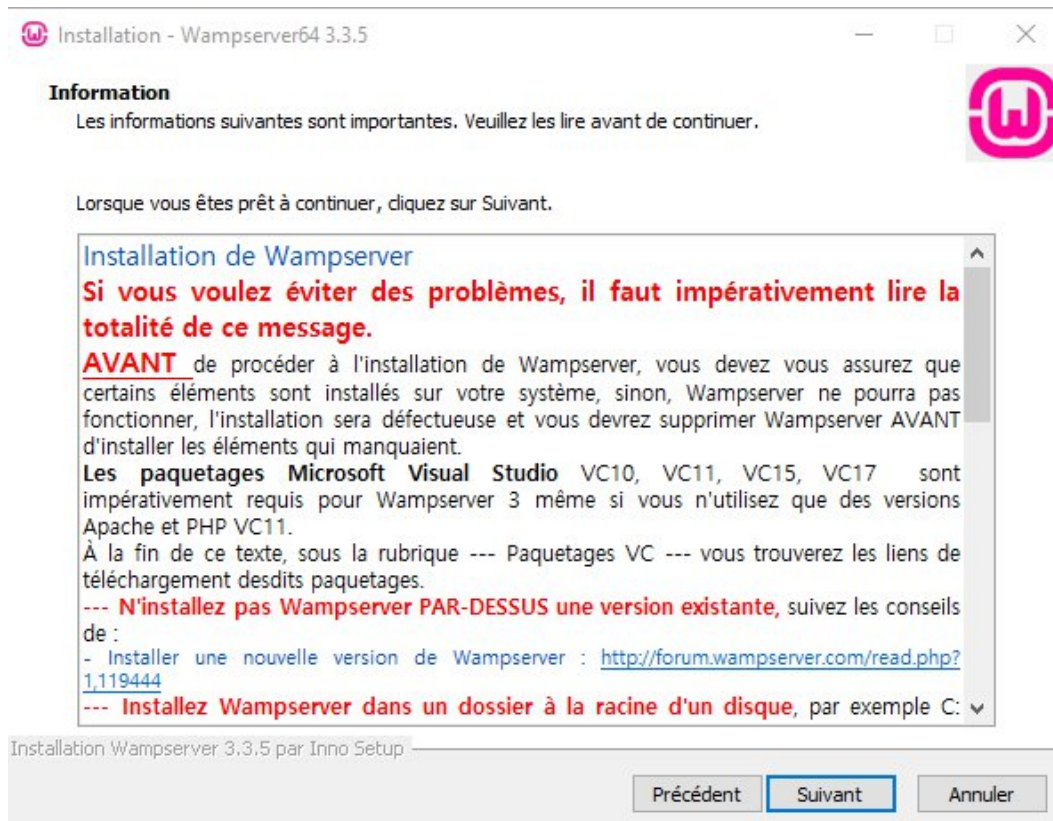
- Installation d'un serveur local :
 - Par WAMP/MAMP

Installation de WAMP (uniquement Windows) :

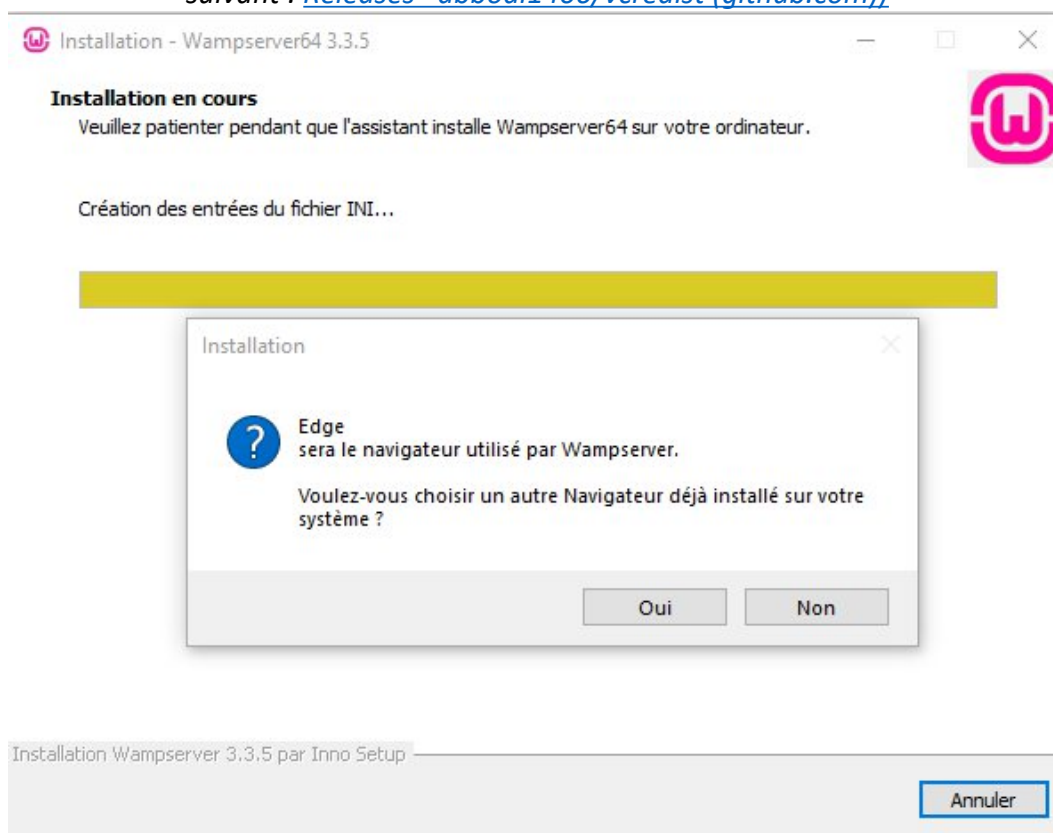
1. Aller sur le site de **WAMP** : <https://www.wampserver.com>
2. Aller sur le menu **Télécharger**
3. Puis cliquer sur le lien **WAMPSERVER 64 BITS**
4. Choisir **passer au téléchargement direct**
5. Cliquer sur **Download Latest Version**
6. Une fois le téléchargement effectué, on peut lancer l'installation :



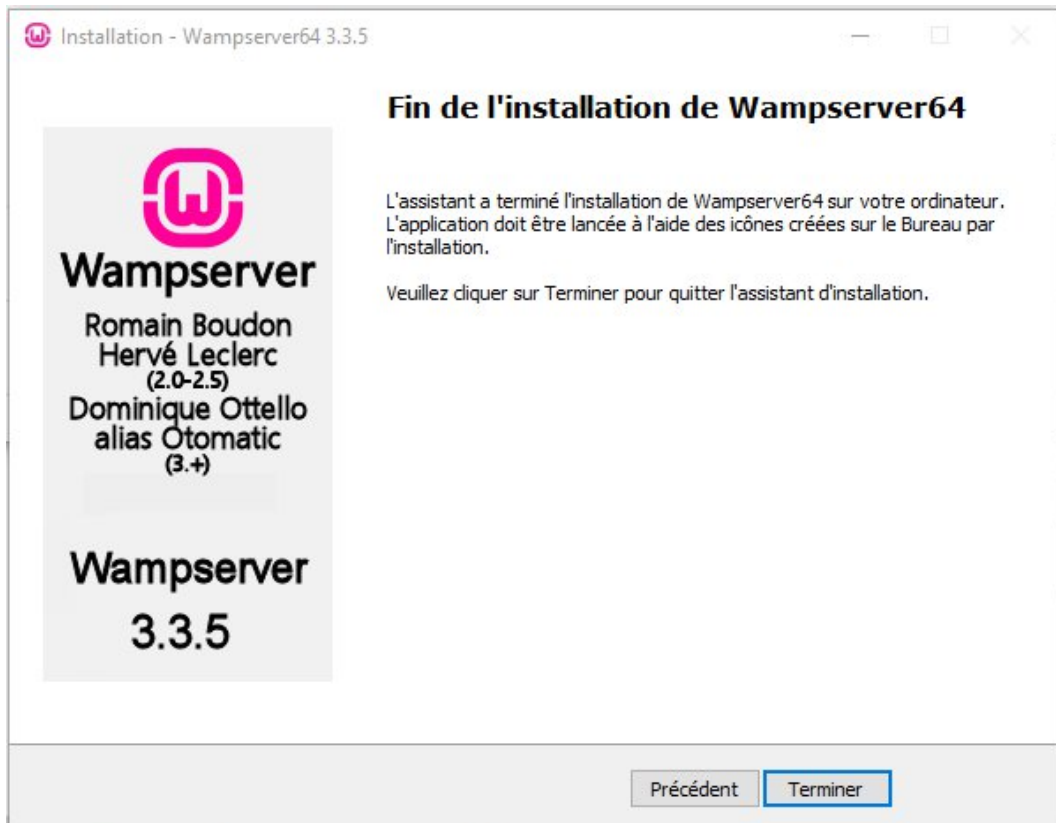
Etape 1 : contrat de licence



Etape 2 : prérequis, les packages Visual Studio (si besoin, il faut les télécharger sur le site suivant : [Releases · abbodi1406/vcredist \(github.com\)](https://github.com/abbodi1406/vcredist))





Etape 3 et 4 : choix du navigateur par défaut et de l'éditeur de texte par défaut

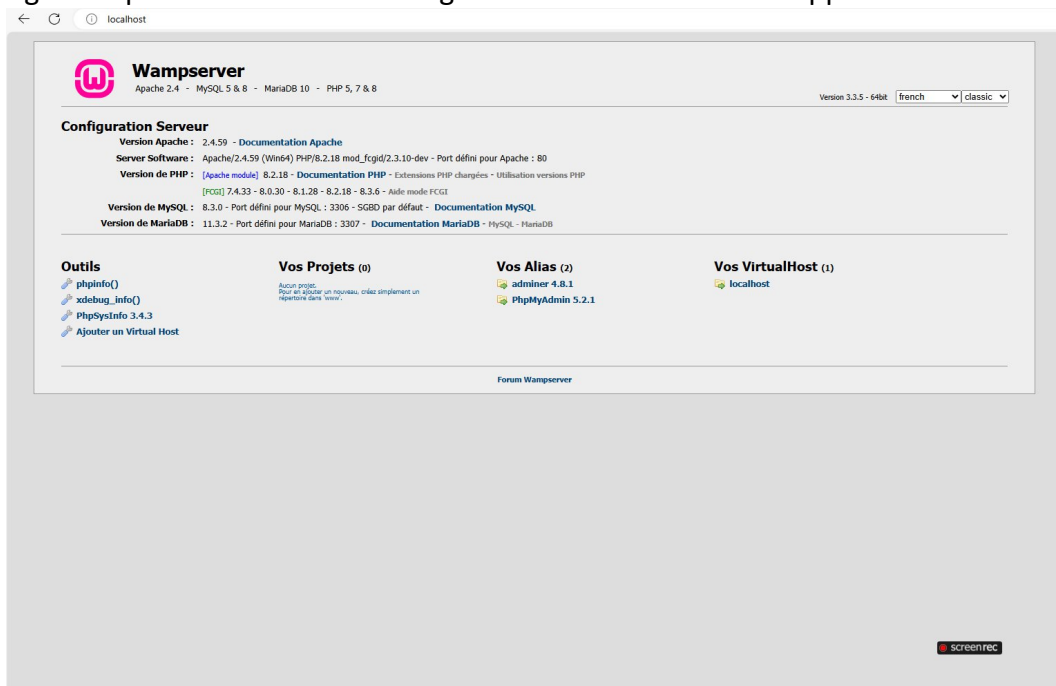


Fin de l'installation

Utilisation de WAMP :

Une fois installé, il faut lancer WampServer par le raccourci présent sur le bureau , une fois lancé, WampServer se retrouve dans la zone de notification .

Un clic gauche permet de lancer la navigation vers le serveur local appelé Localhost :



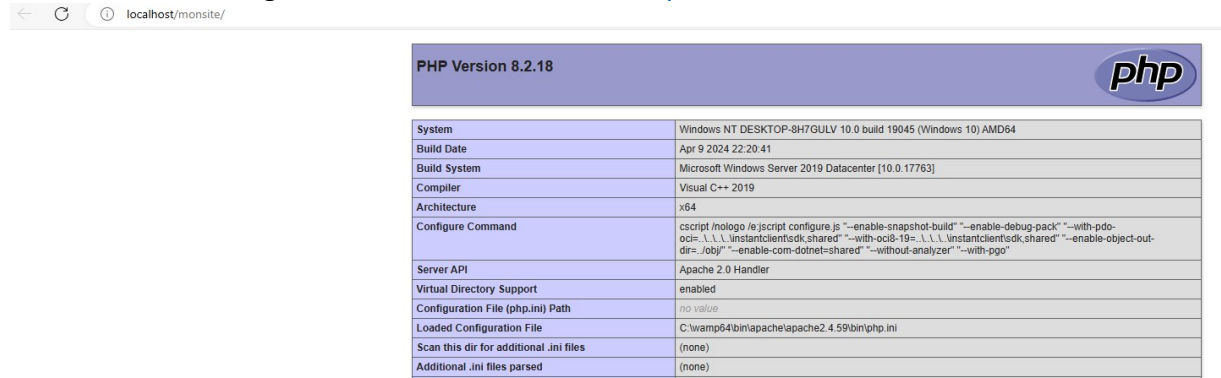
Page d'accueil du serveur local Wamp

Pour créer un projet (un nouveau site), il suffit de créer un dossier dans le répertoire de base de WampServer :

- Aller dans le dossier de WampServer (par défaut : C:\wamp64)
- Aller dans **www**
- Créer un dossier (par ex : monsite)
- Créer un fichier appelé **index.php** avec le contenu suivant

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

- Dans le navigateur, saisir l'url suivante : <http://localhost/monsite>



System	Windows NT DESKTOP-8H7GULV 10.0 build 19045 (Windows 10) AMD64
Build Date	Apr 9 2024 22:20:41
Build System	Microsoft Windows Server 2019 Datacenter [10.0.17763]
Compiler	Visual C++ 2019
Architecture	x64
Configure Command	cmd /c "cd /d %~dp0\php\configure & setlocal & call :__php_build_configure --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:\wamp64\bin\apache\apache2.4.59\bin\php.ini
Scan this dir for additional .ini files	(none)
Additional .ini files parsed	(none)

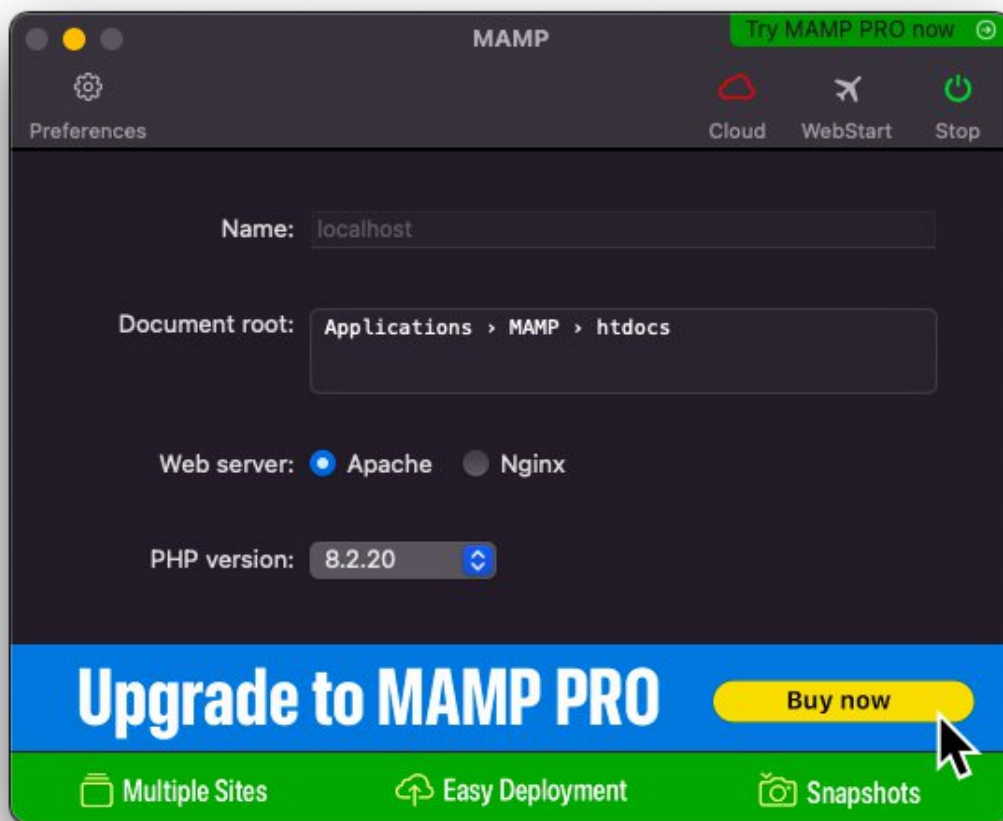
Page du site

Installation de MAMP (au départ prévu pour MacOS, maintenant aussi sur Windows) :

1. Aller sur le site de **MAMP** : <https://www.mamp.info/en/mac/>
2. Aller sur le menu **Download**
3. Une fois le téléchargement effectué, on peut lancer l'installation

Utilisation de MAMP :

Une fois installé, il faut lancer MAMP par le raccourci présent sur le bureau ou dans le LaunchPad de Mac.



Page d'accueil du serveur local MAMP

Pour créer un projet (un nouveau site), il suffit de créer un dossier dans le répertoire de base de MAMP :

- Aller dans le dossier de WampServer (par défaut dans MacOS: Applications MAMP htdocs)
- Aller dans **www**
- Créer un dossier (par ex : monsite)
- Créer un fichier appelé **index.php** avec le contenu suivant

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

- Dans le navigateur, saisir l'url suivante : <http://localhost/monsite>

System	Windows NT DESKTOP-8H7GULV 10.0 build 19045 (Windows 10) AMD64
Build Date	Apr 9 2024 22:20:41
Build System	Microsoft Windows Server 2019 Datacenter [10.0.17763]
Compiler	Visual C++ 2019
Architecture	x64
Configure Command	cmdscript /nologo /e:jsconfig configure.js "--enable-snapshot-build" "--enable-debug-pack" "--with-pdo-oci=\\.\.\.\.instantclient\odk\shared" "--with-oci8-19=\\.\.\.\.instantclient\odk\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:\wamp64\bin\apache\apache2.4.59\bin\php.ini
Scan this dir for additional .ini files	(none)
Additional .ini files parsed	(none)

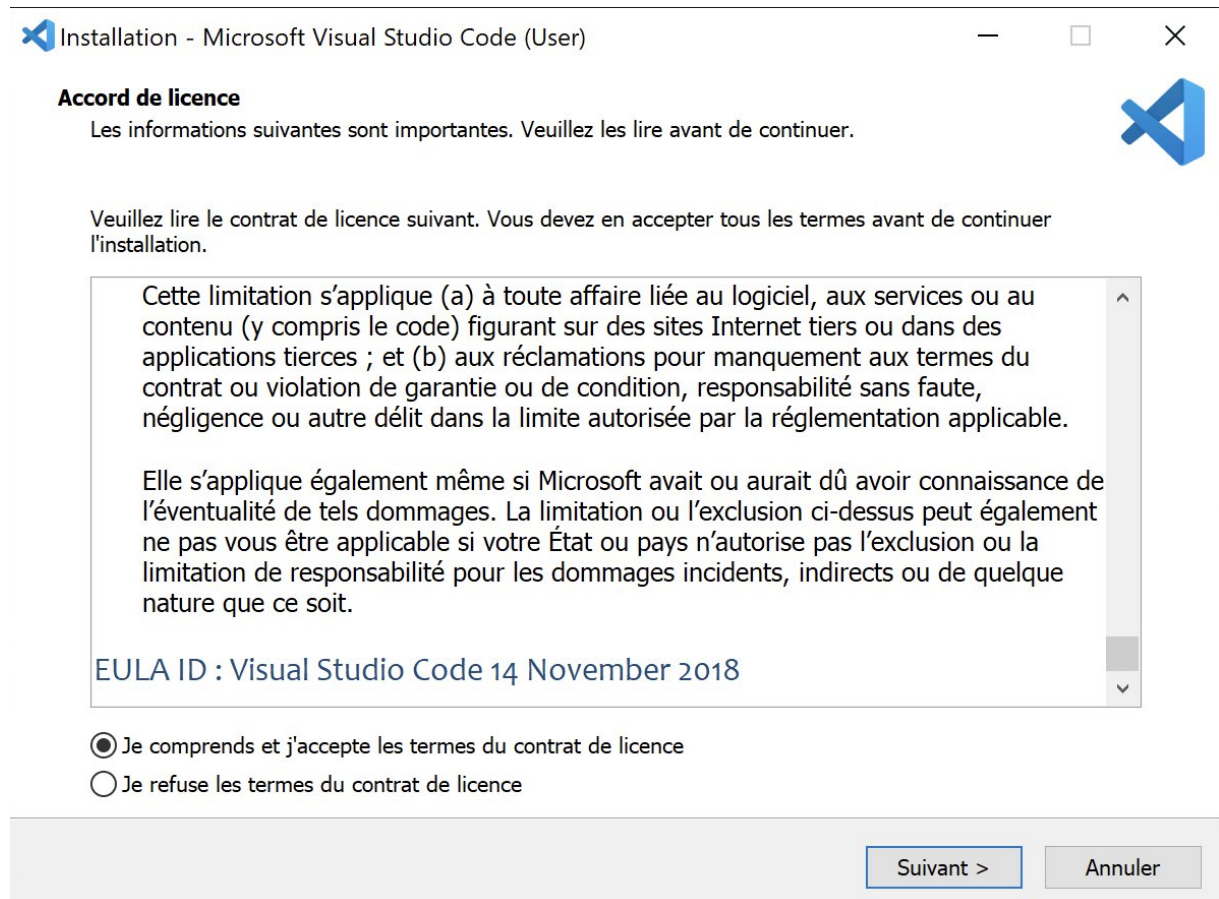
Page du site

- Utilisation d'un IDE : Visual Studio Code
 - Installation

Pour le télécharger, il suffit d'aller à l'adresse suivante : [Download Visual Studio Code - Mac, Linux, Windows](#).

Une fois le téléchargement terminé, double-cliquez sur le fichier d'installation téléchargé pour démarrer le processus d'installation. Voici les différentes étapes :

1. Dans la première étape, vous allez valider les termes du contrat de licence.



The screenshot shows the 'Installation - Microsoft Visual Studio Code (User)' window. The title bar includes the Visual Studio Code icon, the text 'Installation - Microsoft Visual Studio Code (User)', and standard window controls (minimize, maximize, close). The main content area is titled 'Accord de licence' and contains the following text: 'Les informations suivantes sont importantes. Veuillez les lire avant de continuer.' followed by 'Veuillez lire le contrat de licence suivant. Vous devez en accepter tous les termes avant de continuer l'installation.' Below this is a scrollable text box containing the license terms in French, including a limitation on liability. At the bottom of the scrollable area, it says 'EULA ID : Visual Studio Code 14 November 2018'. Below the scrollable area are two radio buttons: 'Je comprends et j'accepte les termes du contrat de licence' (which is selected) and 'Je refuse les termes du contrat de licence'. At the bottom right of the window are two buttons: 'Suivant >' and 'Annuler'.

Validez les termes du contrat de licence

2. Choisissez le dossier dans lequel vous souhaitez installer Visual Studio Code, ou laissez l'emplacement par défaut. Cliquez sur "Suivant".

3. Sélectionnez si vous souhaitez créer un raccourci sur le bureau et/ou dans le menu Démarrer. Cliquez sur "Suivant".

Tâches supplémentaires

Quelles sont les tâches supplémentaires qui doivent être effectuées ?



Sélectionnez les tâches supplémentaires que l'assistant d'installation doit effectuer pendant l'installation de Visual Studio Code, puis cliquez sur Suivant.

Icônes supplémentaires :

☐ Créer une icône sur le Bureau

Autre :

- ☐ Ajouter l'action "Ouvrir avec Code" au menu contextuel de fichier de l'Explorateur Windows
- ☐ Ajouter l'action "Ouvrir avec Code" au menu contextuel de répertoire de l'Explorateur Windows
- ☒ Inscrire Code en tant qu'éditeur pour les types de fichier pris en charge
- ☒ Ajouter à PATH (disponible après le redémarrage)

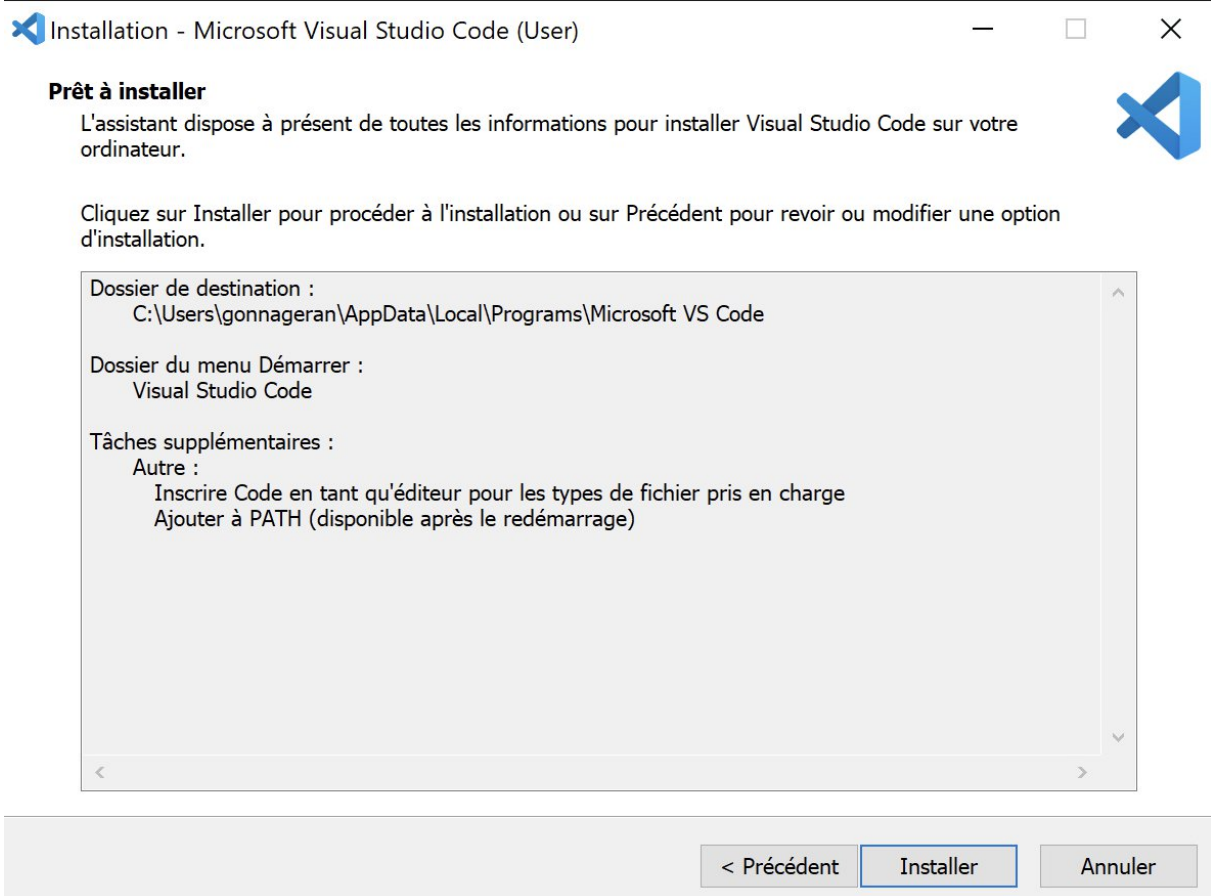
< Précédent

Suivant >

Annuler

Créez un raccourci sur le bureau

4. Cliquez sur "Installer" pour commencer le processus d'installation. Attendez que l'installation soit terminée.

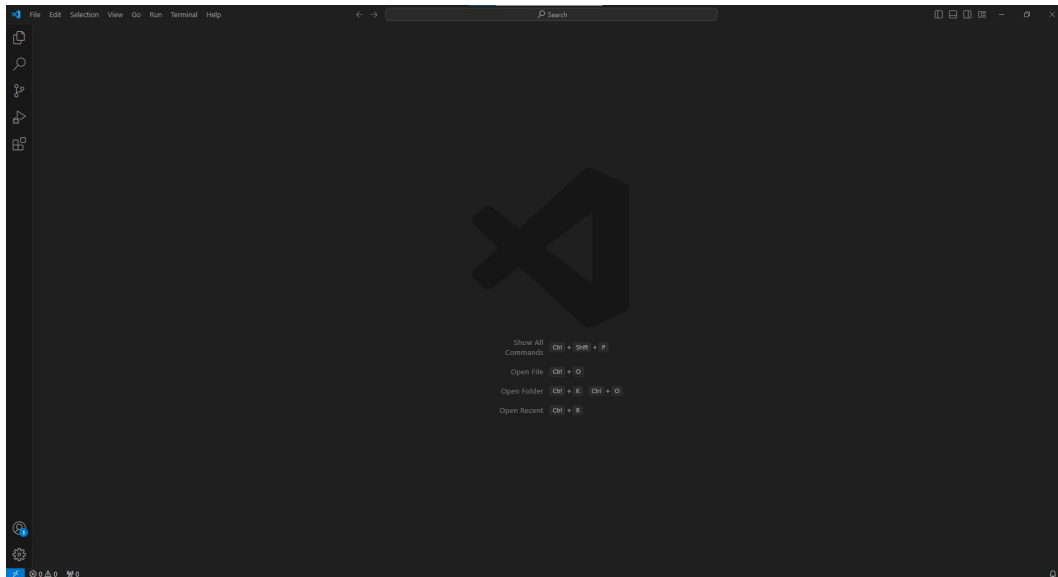


Prêt à installer

5. Une fois l'installation terminée, vous pouvez laisser l'option "Lancer Visual Studio Code" cochée si vous souhaitez ouvrir l'application immédiatement. Cliquez sur "Terminer".

Pour Mac : Le fichier que vous téléchargez est l'application déjà prête à l'emploi. Glissez-la dans votre dossier *Applications* pour la rendre accessible depuis le *LaunchPad*.

Une fois l'installation terminée, vous pouvez lancer Visual Studio Code en recherchant l'application dans le menu de démarrage de votre système d'exploitation ou en double-cliquant sur l'icône sur le bureau, selon votre configuration.



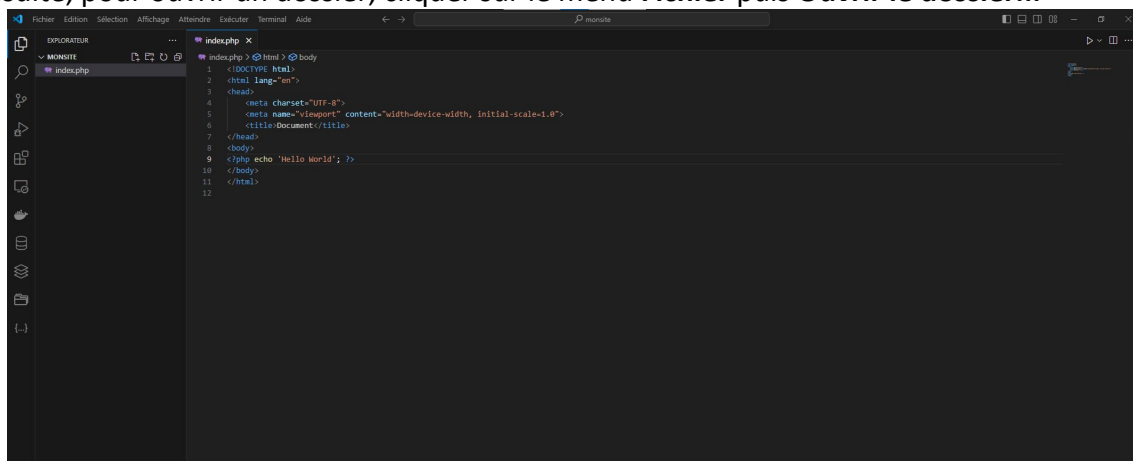
Vue d'ensemble de Visual Studio Code

○ Configuration

La première étape est de mettre Visual Studio Code en français, pour cela :

- Aller dans la palette de commande (dans la barre de recherche en haut, saisir >)
- Saisir **Configure Display Language**
- Choisir **Français**
- Cliquer sur **Restart**

Ensuite, pour ouvrir un dossier, cliquer sur le menu **Fichier** puis **Ouvrir le dossier...**



Dossier ouvert dans Visual Studio Code

Pour créer un fichier dans le projet ouvert il suffit de cliquer sur le menu **Fichier** puis **nouveau Fichier** ou utiliser le raccourci  dans le titre du projet ouvert



Barre de titre du projet dans VS Code

- Extensions indispensables

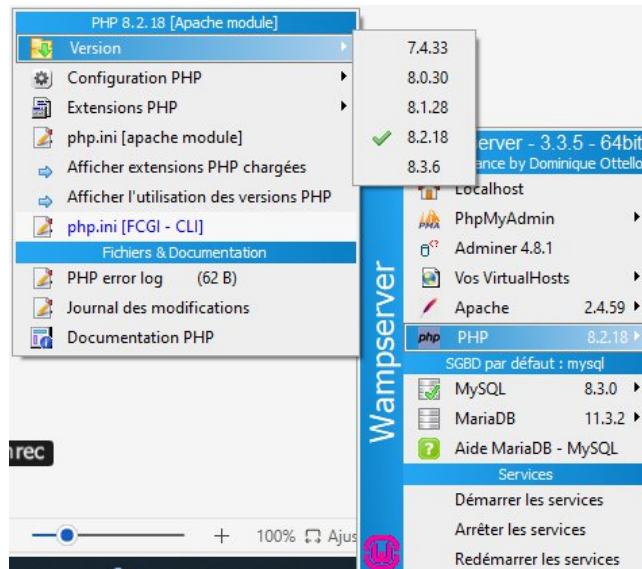
Par défaut, VS Code est installé avec un ensemble de fonctionnalités de base, pour chercher d'autres fonctionnalités ou avoir plus d'options, il suffit d'aller chercher des extensions.

- Général :
 - Project Manager : <https://marketplace.visualstudio.com/items?itemName=alefragnani.project-manager>
 - Path Intellisense : <https://marketplace.visualstudio.com/items?itemName=christian-kohler.path-intellisense>
 - Markdown All in One : <https://marketplace.visualstudio.com/items?itemName=yzhang.markdown-all-in-one>
- Environnement Dev :
 - SFTP : <https://marketplace.visualstudio.com/items?itemName=Natizyskunk.sftp>
 - Remote SSH : <https://marketplace.visualstudio.com/items?itemName=ms-vscode-remote.remote-ssh>
 - Database : <https://marketplace.visualstudio.com/items?itemName=cweijan.vscode-mysql-client2>
 - Docker : <https://marketplace.visualstudio.com/items?itemName=ms-azuretools.vscode-docker>
 - Dev Containers : <https://marketplace.visualstudio.com/items?itemName=ms-vscode-remote.remote-containers>
 - Live Share : <https://marketplace.visualstudio.com/items?itemName=MS-vsliveshare.vsliveshare>
 - Tabnine AI Autocomplete : <https://marketplace.visualstudio.com/items?itemName=TabNine.tabnine-vscode>
 - Auto Rename Tag : <https://marketplace.visualstudio.com/items?itemName=formulahendry.auto-rename-tag>
- PHP :
 - PHP Debug : <https://marketplace.visualstudio.com/items?itemName=xdebug.php-debug>
 - PHP Extension Pack : <https://marketplace.visualstudio.com/items?itemName=xdebug.php-pack>
 - PHP Getters & Setters : <https://marketplace.visualstudio.com/items?itemName=phproberto.vscode-php-getters-setters>
 - PHP Intelephense : <https://marketplace.visualstudio.com/items?itemName=bmewburn.vscode-intelephense-client>
 - PHP IntelliSense : <https://marketplace.visualstudio.com/items?itemName=zobo.php-intellisense>
- JS :
 - Extraits de code JavaScript (ES6) : <https://marketplace.visualstudio.com/items?itemName=xabikos.JavaScriptSnippets>

- ESLint :
<https://marketplace.visualstudio.com/items?itemName=dbaeumer.vscode-eslint>
- TWIG :
 - Twig : <https://marketplace.visualstudio.com/items?itemName=whatwedo.twig>
 - Twig language :
<https://marketplace.visualstudio.com/items?itemName=mblode.twig-language>
 - Twig language 2 :
<https://marketplace.visualstudio.com/items?itemName=mblode.twig-language-2>

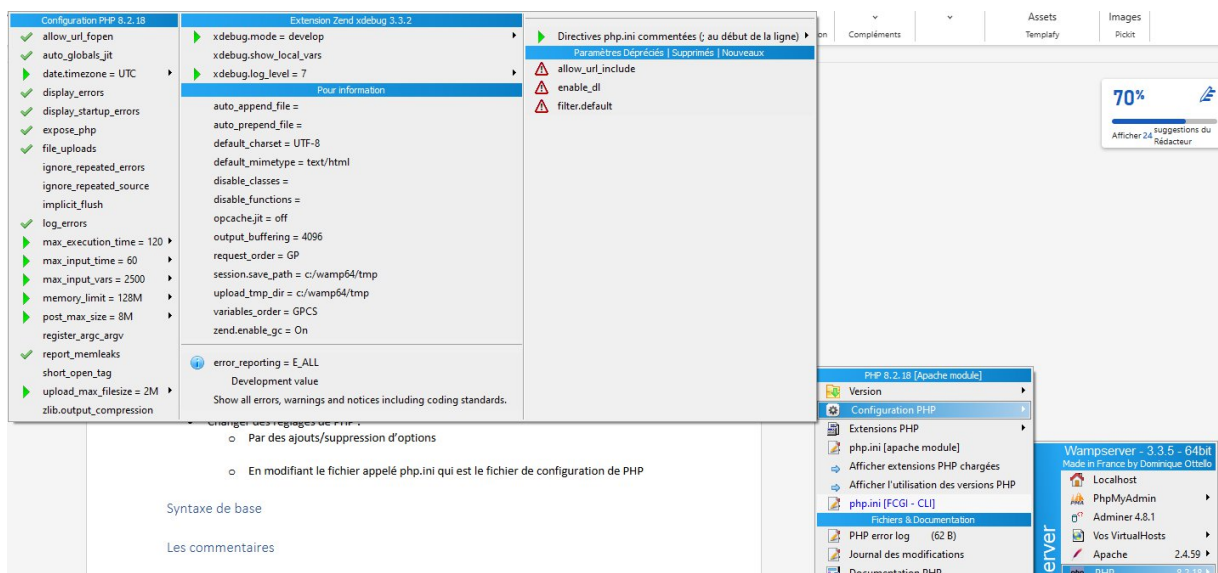
TD 2 : Paramétrage de base concernant PHP et le JavaScript, connexion à la base de données

- Configuration du serveur :
 - WAMP :
 - Changer la version de PHP



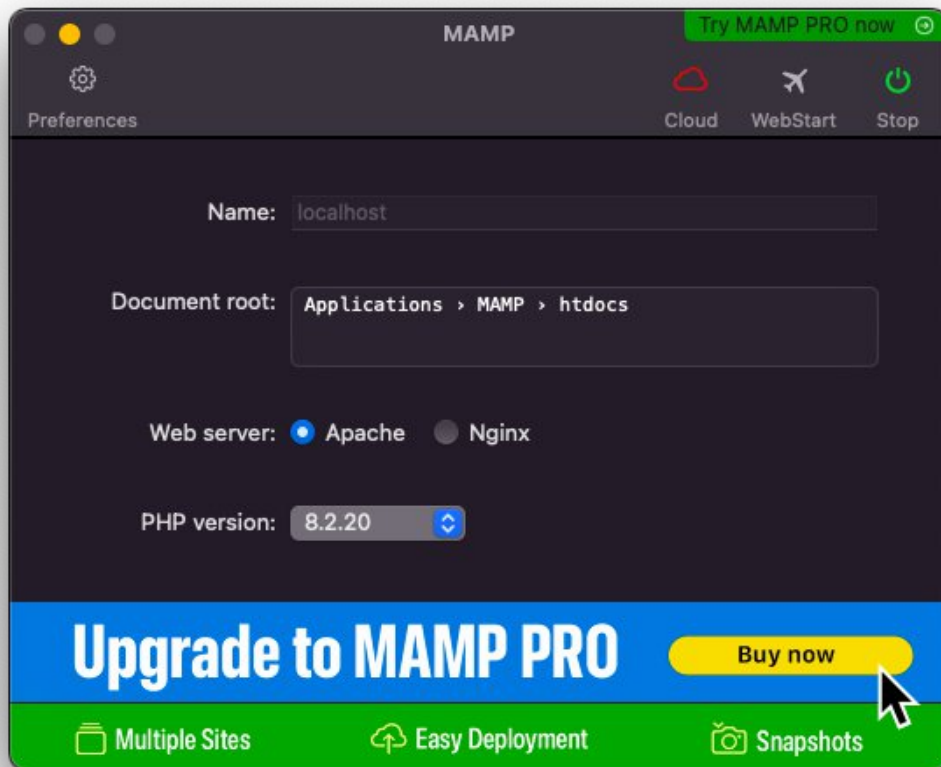
Liste des versions de PHP

- Par des ajouts/suppression d'options liés à PHP (change le fichier php.ini)

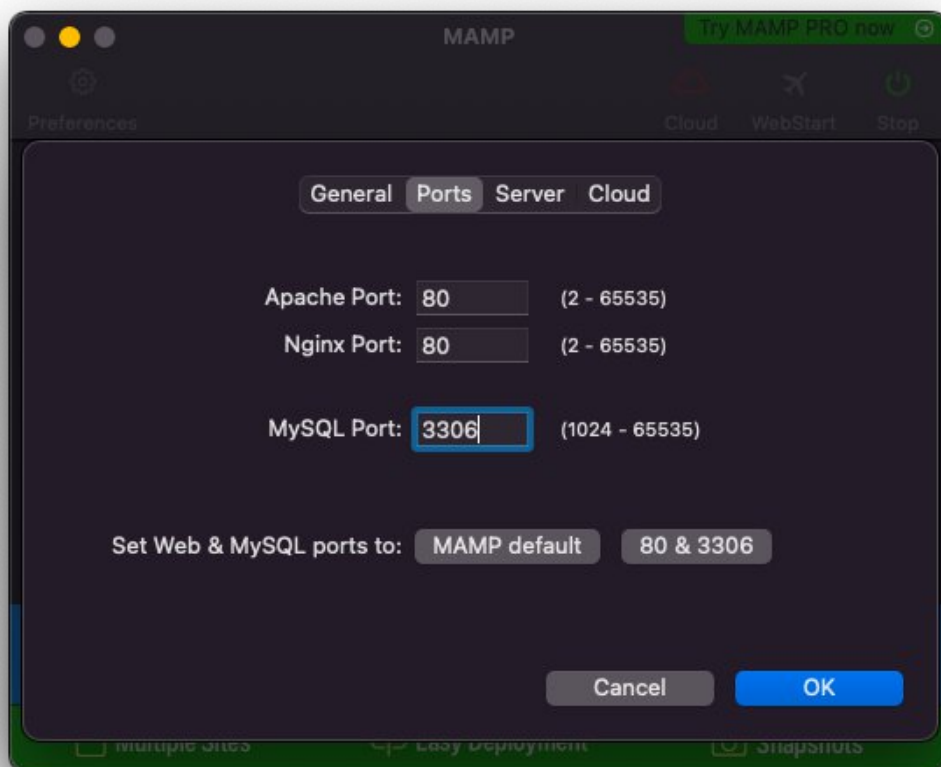


Configuration de PHP

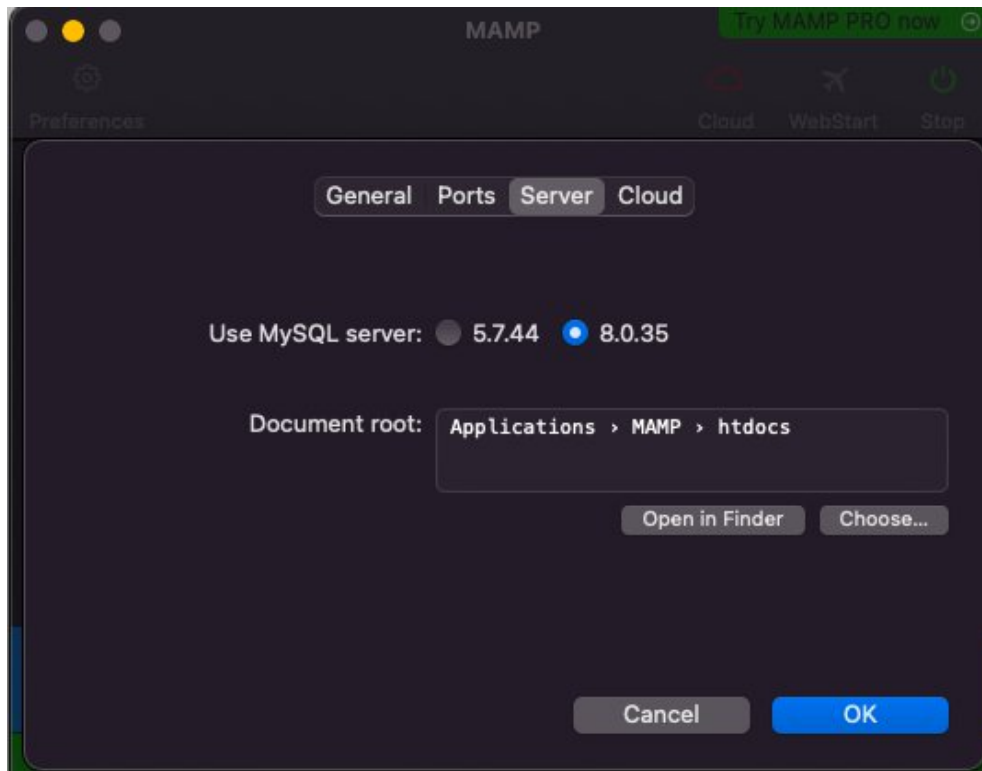
- MAMP :



Choix du PHP sur la page d'accueil




Définition des ports



Choix de la version du serveur de base de données

○ Les DevContainers (conteneur de développement) de VSCode
C'est tout nouveau et tout beau. Cela permet de créer un serveur local intra VS Code via Docker. Il faut donc disposer de Docker au préalable.

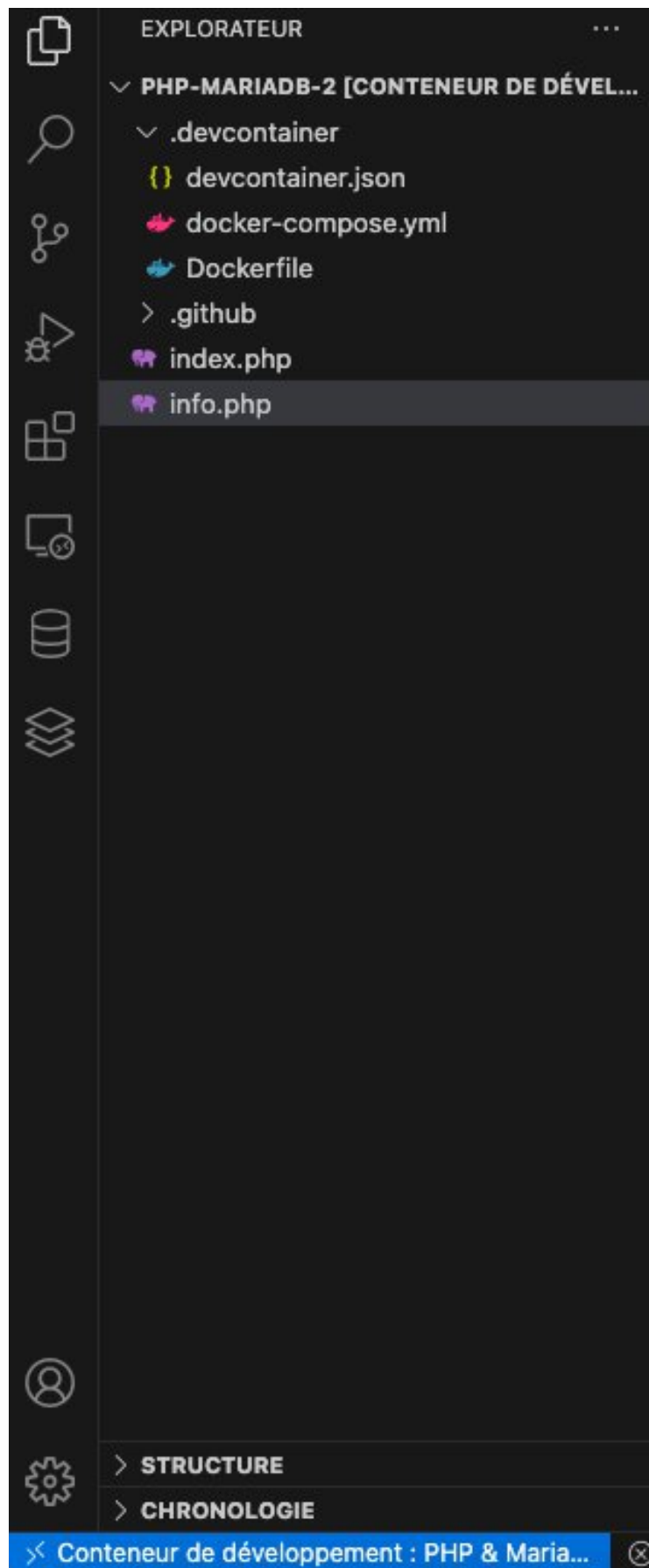
- Installation de Docker :
 - Sur Windows : <https://docs.docker.com/desktop/install/windows-install/>
 - Sur MacOS : <https://docs.docker.com/desktop/install/mac-install/>
- Installation (si besoin) de l'extension Dev Containers dans VSCode : <https://marketplace.visualstudio.com/items?itemName=ms-vscode-remote.remote-containers>
- Création d'un container PHP MariaDB

Il faut d'abord aller dans les connexions à distance :  (en bas à gauche de VSCode)

Choisir **Nouveau conteneur de développement...**

Sélectionner le modèle de conteneur PHP & MariaDB

Le conteneur s'installe, jusqu'à ce qu'il soit chargé



Conteneur chargé

- Configuration et tests

Modifier le fichier de configuration (**devcontainer.json**) comme suit :

```
// For format details, see https://aka.ms/devcontainer.json. For config options, see the
// README at: https://github.com/devcontainers/templates/tree/main/src/php-mariadb
{
  "name": "PHP & MariaDB",
  "dockerComposeFile": "docker-compose.yml",
  "service": "app",
  "workspaceFolder": "/workspaces/${localWorkspaceFolderBasename}",

  // Features to add to the dev container. More info: https://containers.dev/features.
  // "features": {},

  // For use with PHP or Apache (e.g.php -S localhost:8080 or apache2ctl start)
  "forwardPorts": [8080, 3306],

  // Use 'postCreateCommand' to run commands after the container is created.
  "postCreateCommand": "sudo chmod a+x \"$(pwd)\" && sudo rm -rf /var/www/html && sudo ln -s \"$(pwd)\" /var/www/html"

  // Configure tool-specific properties.
  // "customizations": {},

  // Uncomment to connect as root instead. More info: https://aka.ms/dev-containers-non-root.
  // "remoteUser": "root"
}
```

Puis pour lancer le service apache (serveur Web), il faut aller sur le Terminal (menu Affichage puis Terminal), saisir la commande : **apache2ctl start**.

Créer un fichier pour tester le fonctionnement, par exemple, **index.php** avec le contenu suivant :

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

Tester dans un navigateur l'url : <http://localhost:8080>.

- Configuration de PHP :

Afin de pouvoir agir sur de nombreux paramètres (poids de fichier à importer, temps limite d'exécution d'un script, affichage d'erreur dans le navigateur, ...), la configuration de PHP est une étape indispensable.

Dans **WAMP**, certains réglages de PHP peuvent être faits comme vu précédemment. Ceux-ci modifient le fichier **php.ini**.

Ce fichier peut être modifier directement, pour cela il faut connaitre son emplacement :

- Dans le cas de WAMP :

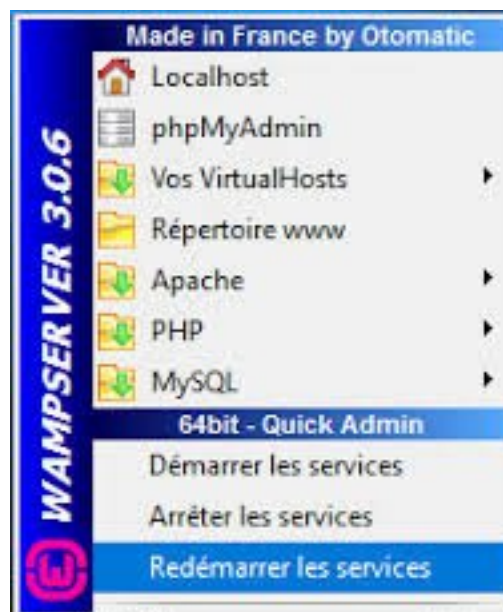
- Dans le cas de MAMP : /Applications/MAMP/bin/php/php8.2.20/conf/php.ini

Quelques réglages php à connaître et à modifier si besoin :

- **memory_limit** : définit le poids maximum pris en mémoire pour l'exécution d'un script php
- **max_execution_time** : définit la durée maximum d'exécution d'un script php
- **error_reporting** : définit les catégories d'erreurs qui seront affichées et/ou mis dans le log (journal), par défaut E_ALL (toutes les erreurs)
- **display_errors** : définit si les erreurs php sont affichées sur la page affichée sur le navigateur, peut-être On ou Off
- **log_errors** : définit si les erreurs php sont dans un fichier de log (journal), peut-être On ou Off
- **error_log** : emplacement du fichier permettant de visualiser les erreurs php
- **post_max_size** : définit le poids maximum d'un chargement en POST
- **upload_max_filesize** : définit le poids maximum d'un fichier chargé

Important : à chaque modification du fichier php.ini, il faut redémarrer le serveur.

Dans **WAMP** : dans la zone de notification, faire un clic gauche puis **Redémarrer les services**



Vue de WAMP

Dans **MAMP** : dans l'interface de MAMP, il faut cliquer sur **Stop** puis sur **Start**



Vue de MAMP

- Configuration et saisie de JavaScript :

- Par la console du navigateur

Tests avec : alert, console.log, let et quelques d'instructions.

Créer les variables suivantes (avec let)

- num qui contient le nombre 255
- txt qui contient la chaîne de caractère 255
- bin qui contient la valeur booléenne vraie
- flt qui contient la valeur 7,23

Reprenez la déclaration des variables précédentes et :

- Appliquez à num l'opérateur d'incrément
- Concaténez à la chaîne txt la chaîne " est ici une chaîne de caractères"
- Inversez la valeur logique de bin
- Ajoutez à flt la valeur contenue dans num

Grâce à la fonction prompt(), demander l'âge de l'utilisateur, puis à l'aide d'une condition, déterminer et afficher via une boîte de dialogue si l'utilisateur est mineur ou majeur. On considère qu'un individu est majeur à partir de 18 ans.

- Par <https://playcode.io/>

Tests avec du HTML en même temps pour voir les éléments du DOM/BOM

Index.html

```
<!DOCTYPE html>
<html lang="en">
  <head>
    <title>Bonjour</title>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <link rel="stylesheet" href="src/style.css">
  </head>
  <body>
    <h1 id="header"></h1>
    <br>
    <div id="main">
      <ul>
        <li>Premier</li>
        <li>Deuxième</li>
      </ul>
    </div>
    <script src="src/script.js"></script>
  </body>
</html>
```

Script.js

```
const message = 'Je suis là' // Try edit me

// Update header text
document.querySelector('#header').innerHTML = message

alert(document.title);

// Log to console
console.log(message)

let main = document.getElementById("main");

let liElements = main.getElementsByTagName("li");

for(const li of liElements) {
  console.log(li.textContent)
}
```

- Configuration en PHP d'une base de données :
 - Créer une base de données via VS Code (extension Database Client <https://marketplace.visualstudio.com/items?itemName=cweijan.vscodex-database-client2>)
 - **Nommer la base** : monsite
 - **Créer un dossier dans le dossier de WAMP (www) ou MAMP (htdocs)** : monsite
 - **L'ouvrir** dans VSCode
 - **Créer** le fichier index.php, avec le contenu suivant :

```
<?php
try
{
    $mysqlClient = new PDO('mysql:host=localhost;dbname=monsite;charset=utf8', 'root', '');
    echo "la connection est faite !" ;
}
catch (Exception $e)
{
    die('Erreur : ' . $e->getMessage());
}
?>
```

- **Tester** le site dans le navigateur
- Premier site en PHP version MVC :

Fichier index.php

```
<!DOCTYPE html>
<html lang="fr">
<head>
  <meta charset="UTF-8">
  <meta name="viewport" content="width=device-width, initial-scale=1.0">
  <title>Bienvenue</title>
  <nav>
    <ul>
      <li><a href="#">Accueil</a></li>
      <li><a href="#">Contact</a></li>
    </ul>
  </nav>
</head>
<?php
  try
  {
    // Connexion à la base de données
    $host = "localhost";
    $dbname = "monmvc";
    $user = "root";
    $password = "root";
    $pdo = new PDO("mysql:host=$host;dbname=$dbname", $user, $password);
    //echo "la connexion est faite !";
  }
  catch (Exception $e)
  {
    die('Erreur : ' . $e->getMessage());
  }

  // Connexion à la base de données
  $qr = $pdo->query('SELECT * FROM recipes');
  $qr->execute();
  $recipes = $qr->fetchAll(PDO::FETCH_ASSOC);
?>

<body>
  <h1>Bienvenue sur mon site !</h1>
  <p>
    <?php
      $date = date('d/m/Y');
      $heure = date('H:i:s');
      echo "Nous sommes le {$date}<br>"; // date du jour
      echo 'Il est ' . $heure;
    ?>
  </p>
  <table border="1">
    <thead>
      <th>Nom</th>
      <th>Mail de l'auteur</th>
    </thead>
  </table>
```



```

</thead>
<?php foreach($recipies as $recipe) : ?>
    <tr>
        <td><?= $recipe['nom'] ?></td>
        <td><?= $recipe['email'] ?></td>
    </tr>
<?php endforeach; ?>
</table>
</body>

<footer>
    <p>&copy; 2024 - Tous droits réservés</p>
</footer>
</html>

```

Tester le site en l'ouvrant dans le navigateur

Effectuer le découpage pour appliquer le modèle MVC :

- **Créer les dossiers** : controllers, models et views
- **Créer les fichiers** :
 - header.php, home.php et footer.php dans views
 - homeController.php dans controllers
 - db.php dans models
- **Modifier** le contenu de chaque fichier comme suit :

Fichier header.php :

```

<!DOCTYPE html>
<html lang="fr">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Bienvenue</title>
    <nav>
        <ul>
            <li><a href="#">Accueil</a></li>
            <li><a href="#">Contact</a></li>
        </ul>
    </nav>
</head>

```

Fichier footer.php

```

<footer>
    <p>&copy; 2024 - Tous droits réservés</p>
</footer>
</html>

```

Fichier home.php

```

<body>
  <h1>Bienvenue sur mon site !</h1>
  <p>
    <?php
      $date = date('d/m/Y');
      $heure = date('H:i:s');
      echo "Nous sommes le {$date}<br>"; // date du jour
      echo 'Il est ' . $heure;
    ?>
  </p>
  <table border="1">
    <thead>
      <th>Nom</th>
      <th>Mail de l'auteur</th>
    </thead>
    <?php foreach($recipes as $recipe) : ?>
      <tr>
        <td><?= $recipe['nom'] ?></td>
        <td><?= $recipe['email'] ?></td>
      </tr>
    <?php endforeach; ?>
  </table>
</body>

```

Fichier homeController.php

```

<?php

// Connexion à la base de données
$qqr = $pdo->query('SELECT * FROM recipes');
$qqr->execute();
$recipes = $qqr->fetchAll(PDO::FETCH_ASSOC);

require_once('./views/home.php');

?>

```

Fichier db.php

```

<?php
try
{
  // Connexion à la base de données
  $host = "localhost";
  $dbname = "monmvc";
  $user = "root";
  $password = "root";
  $pdo = new PDO("mysql:host=$host;dbname=$dbname", $user, $password);
  //echo "la connexion est faite !";
}
catch (Exception $e)

```

```
{  
    die('Erreur : ' . $e->getMessage());  
}  
  
?>
```

Fichier index.php

```
<?php  
  
    // Connexion à la base de données  
    require_once(__DIR__.DIRECTORY_SEPARATOR.'models'.DIRECTORY_SEPARATOR.'db.php');  
  
    // Affichage du header  
    require_once(__DIR__.DIRECTORY_SEPARATOR.'views'.DIRECTORY_SEPARATOR.'header.php');  
  
    // Affichage du controleur d'accueil  
  
require_once(__DIR__.DIRECTORY_SEPARATOR.'controllers'.DIRECTORY_SEPARATOR.'homeController.php');  
  
    // Affichage du footer  
    require_once(__DIR__.DIRECTORY_SEPARATOR.'views'.DIRECTORY_SEPARATOR.'footer.php');  
  
?>
```