# **Installation de XDebug : debugging en PHP**

1. Vérifiez si vous avez déjà le fichier du module XDebug dans votre disque : allez dans **xampp/php/ext** et cherchez le fichier **php\_xdebug.dll**

Si c’est le cas, vous avez déjà le module et il reste qu’à l’activer. Passez alors directement au point 3 de ce tutoriel

1. **Téléchargez XDebug** de xDebug.org (le fichier .dll de la version qui vous correspond, 64 bits !) et copiez-le dans le dossier **modules** d’Apache (dans XAmpp)

**Pour connaitre votre version du php :** Créez un fichier **info.php** contenant un appel à **phpinfo()** et lancez-le. Notez quelle est votre version de PHP

En Windows il s’agit d’un fichier dll qu’on doit renommer à **php\_xdebug.dll** et le placer dans **xampp/php/ext**

1. **Editez le fichier php.ini (xampp/php)**

Créez la section **[XDebug]** contenant les lignes suivantes (à la fin du fichier php.ini, par exemple):

**[XDebug]**

**zend\_extension = "C:\xampp\php\ext\php\_xdebug.dll"**

**xdebug.remote\_host = "127.0.0.1"**

**xdebug.remote\_enable = 1**

**xdebug.remote\_handler = "dbgp"**

Note : cette section peut exister déjà. Si c’est le cas, mettez-la à jour. Rajustez les chemins selon vos besoins

1. Créez un fichier **info.php** dans htdocs. Créez une page web de base et rajoutez la ce code dans le body:

<?php

    phpinfo();

?>

Redémarrez **apache.** Vérifiez que l’installation a été réalisée: relancez la page info.php et cherchez la section **xdebug**. Le module doit être marqué comme **enabled**.

Vous pouvez aussi faire un bête var\_dump d’un array et observer que le format d’affichage a changé…

1. Vous pouvez debugger PHP dans VS Code. Installez l'extension [PHP Debug](https://marketplace.visualstudio.com/items?itemName=felixfbecker.php-debug) dans Visual Studio. Puis rajoutez cette ligne à la section **[XDebug]** dans **php.ini** (dossier xampp\php) :

**xdebug.remote\_autostart = 1**

Cette ligne lancera le debugger quand on rechargera la page.

Pour tester le bon fonctionnement :

1. Redemarrez VS Code et Apache
2. Lancez le debugger et choisissez PHP. Il y a une configuration par défaut dans le debugger qui doit fonctionner (cliquez sur l'icône de debugging en bas à gauche de la fenêtre et puis sur "Add Configuration🡪PHP")
3. Créez de breakpoints
4. Ouvrez le script dans le navigateur
5. Si vous utilisez NetBeans, toutes les options de Debug devraient être activées. Vous pouvez utiliser **Debug File** si vous voulez lancer un fichier PHP en particulier. Pour le reste, ça fonctionne comme n’importe quel autre Debugger (breakpoints, watches etc…).

Raccourcis utiles :

* **Activer/desactiver un breakpoint** dans une ligne:
* **Step Into: F7** – Lancer la ligne suivante. S'il s'agit d'un appel à une fonction, lancer ligne par ligne le code de la fonction/méthode
* **Step Over: F8** – Même chose mais on ne visualisera pas l'exécution de chaque ligne de la fonction/méthode, celle-ci sera lancé comme un bloc unitaires
* **Step Out: CTRL + F7** – Si on se trouve dans une fonction, lancer le reste de lignes de la fonction et continuer l'exécution depuis l'appel. Si on se trouve dans le code principal, continuer l'exécution jusqu'au prochain breakpoint

Conseil : N’oubliez pas de modifier l’URL de votre projet dans **Project properties (Clic droit sur le projet) 🡪Run Configuration** pour pouvoir lancer directement votre projet depuis NetBeans

1. Si vous voulez lancer le debugger pour un ensemble de pages d'un projet (ex: une page PHP qui contient un formulaire qui appelle une autre page PHP) vous devez établir l'URL de la première page que vous voulez ouvrir dans index file. Puis vous devez cliquer sur CTRL-F5 (Debug Project)

