

# Debug en PHP

---

# Uso actual vs Realidad

- echo “...”
- var\_dump()
- die()



## Debug en un mundo complicado

- Similar al que usamos para JavaScript
- Permite recorrer el código paso a paso
- Ver el estado de cada variable en cualquier instante
- Ingresar a un método si es necesario

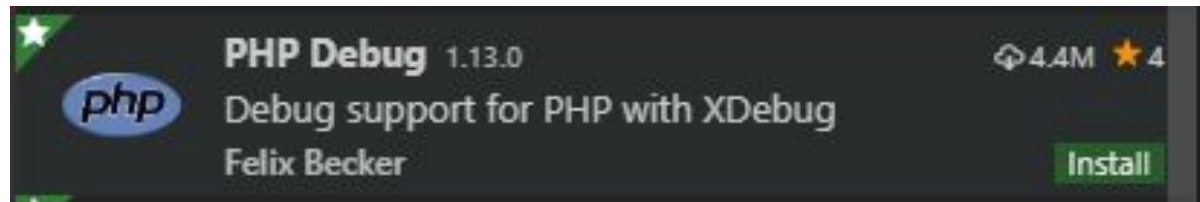
# Debug en un mundo complicado

---

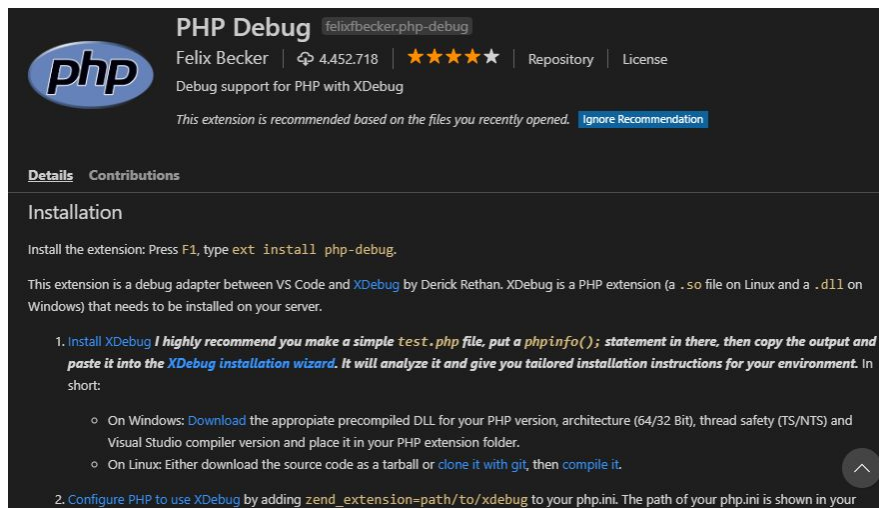
- Similar al que usamos para JavaScript
- Permite recorrer el código paso a paso
- Ver el estado de cada variable en cualquier instante
- Ingresar a un método si es necesario

# Como instalar XDebug en VS Code

## 1. Instalar Extension “PHP Debug” en VS Code



## 2. Seguir los pasos recomendados por la Extensions



# Pasos de la Extension PHP Debug

1. Crear archivo “**test.php**” y escribir **phpinfo()**;
2. Ingresar a [XDebug installation wizard](#)
3. Copiar toda lo que imprime phpinfo() en el cuadro de dialogo

localhost/test.php	
XML	Rig Bakken, Thig G. Amken, Sterling Hughes
XML Writer	Rob Richards, Pierre-Alain Joye
XSL	Christian Stocker, Rob Richards
Zip	Pierre-Alain Joye, Remi Collet
File	Rasmus Lerdorf, Stefan Roderich, Zeev Suraski, Jade Nicoletti, Michael Wallner
PHP Documentation	
Authors	Mehdi Achour, Friedhelm Betz, Antony Dougal, Nuno Lopes, Hannes Magnusson, Philip Olson, Georg Richter, Damien Sogay, Jakub Vrana, Adam Harvey
Editor	Peter Couburn
User Note Maintainers	Daniel P. Brown, Thiago Henrique Poide
Other Contributors	Previously active authors, editors and other contributors are listed in the manual.
PHP Quality Assurance Team	
Ira Alshenetsky, Joerg Behrens, Antony Dougal, Stefan Esser, Moryoshi Koizumi, Magnus Maatta, Sebastian Mohr, Derrick Rothens, Melynn Sopacua, Pierre-Alain Joye, Dmitry Slogov, Felipe Pena, David Soria Parra, Stanislav Melyshev, Julien Pauli, Stephen Zarkos, Anatol Belski, Remi Collet, Ferenc Kovacs	
Websites and Infrastructure team	
PHP Websites Team	Rasmus Lerdorf, Hannes Magnusson, Philip Olson, Lukas Kahwe Smith, Pierre-Alain Joye, Kalle Sommer Nielsen, Peter Couburn, Adam Harvey, Ferenc Kovacs, Levi Morrison
User Note Maintainers	Daniel P. Brown, Daniel P. Brown
Network Infrastructure	Daniel P. Brown
Windows Infrastructure	Alan Schoenmaler



← → ↻ xdebug.org/wizard.php

DDEBUG EXTENSION FOR PHP | DOCUMENTATION | INSTALLATION

[home](#) | [updates](#) | [download](#) | [documentation](#) | [contributing](#) | [license](#) | [support](#) | [issue tracker](#)

TAILORED INSTALLATION INSTRUCTIONS

This page helps you finding which file to download, and how to configure PHP to get Xdebug running. Please paste the **full** output of phpinfo() (either a copy & paste of the HTML version, the HTML source or `php -i` output) and submit the form to receive tailored download and installation instructions.

XSL

Christian Stocker, Rob Richards

Zip

Pierre-Alain Joye, Remi Collet

File

Rasmus Lerdorf, Stefan Roderich, Zeev Suraski, Jade Nicoletti, Michael Wallner

PHP Documentation

Authors Mehdi Achour, Friedhelm Betz, Antony Dougal, Nuno Lopes, Hannes Magnusson, Philip Olson, Georg Richter, Damien Sogay, Jakub Vrana, Adam Harvey

Editor

Peter Couburn

User Note Maintainers

Daniel P. Brown, Thiago Henrique Poide

Other Contributors

Previously active authors, editors and other contributors are listed in the manual.

PHP Quality Assurance Team

Ira Alshenetsky, Joerg Behrens, Antony Dougal, Stefan Esser, Moryoshi Koizumi, Magnus Maatta, Sebastian Mohr, Derrick Rothens, Melynn Sopacua, Pierre-Alain Joye, Dmitry Slogov, Felipe Pena, David Soria Parra, Stanislav Melyshev, Julien Pauli, Stephen Zarkos, Anatol Belski, Remi Collet, Ferenc Kovacs

Websites and Infrastructure team

PHP Websites Team Rasmus Lerdorf, Hannes Magnusson, Philip Olson, Lukas Kahwe Smith, Pierre-Alain Joye, Kalle Sommer Nielsen, Peter Couburn, Adam Harvey, Ferenc Kovacs, Levi Morrison

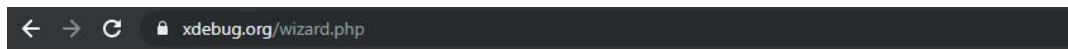
If you like Xdebug, please consider giving a [donation](#).

BECOME A PATRON

# Pasos de la Extension PHP Debug

4. Hacer click en el boton “Analyse my phpinfo() output”

5. Seguir los pasos indicados por el wizard



## SUMMARY

- **Xdebug installed:** no
- **Server API:** Apache 2.0 Handler
- **Windows:** yes - Compiler: MS VC15 - Architecture: x64
- **Zend Server:** no
- **PHP Version:** 7.3.8
- **Zend API nr:** 320180731
- **PHP API nr:** 20180731
- **Debug Build:** no
- **Thread Safe Build:** yes
- **OPcache Loaded:** no
- **Configuration File Path:** C:\WINDOWS
- **Configuration File:** C:\xampp\php\php.ini
- **Extensions directory:** C:\xampp\php\ext

## INSTRUCTIONS

1. Download [php\\_xdebug-2.7.2-7.3-vc15-x86\\_64.dll](#)
2. Move the downloaded file to C:\xampp\php\ext
3. Edit C:\xampp\php\php.ini and add the line  
`zend_extension = C:\xampp\php\ext\php_xdebug-2.7.2-7.3-vc15-x86_64.dll`
4. Restart the webserver

# Pasos del Wizard de XDebug

1. Descargar [php\\_xdebug-2.7.2-7.3-vc15-x86\\_64.dll](#)
  - El nombre el archivo va a cambiar dependiendo la version de PHP
  - En caso de utilizar Linux sera otro archivo
2. Copiar el archivo descargado a “C:\xampp\php\ext”
3. Editar “C:\xampp\php\php.ini” y agregar las lineas
  - [XDebug]
  - xdebug.remote\_enable = 1
  - xdebug.remote\_autostart = 1
  - zend\_extension = C:\xampp\php\ext\php\_xdebug-2.7.2-7.3-vc15-x86\_64.dll

```
[XDebug]
xdebug.remote_enable = 1
xdebug.remote_autostart = 1
zend_extension = C:\xampp\php\ext\php_xdebug-2.7.2-7.3-vc15-x86_64.dll
```

4. Reiniciar el Webserver

# Verificacion de XDebug

1. Ejecutar nuevamente el archivo “**test.php**”
2. Dentro del **phpinfo()** deberia aparecer informacion de XDebug

## xdebug

xdebug support	enabled
Version	2.7.2
IDE Key	jromero

Support **Xdebug** on Patreon

**BECOME A PATRON**

## Supported protocols

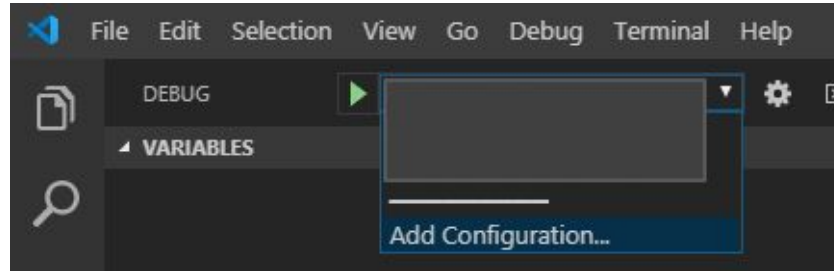
DBGp - Common DeBuGger Protocol

Directive	Local Value	Master Value
<b>xdebug</b> .auto_trace	Off	Off
<b>xdebug</b> .cli_color	0	0
<b>xdebug</b> .collect_assignments	Off	Off
<b>xdebug</b> .collect_includes	On	On
<b>xdebug</b> .collect_params	0	0
<b>xdebug</b> .collect_return	Off	Off
<b>xdebug</b> .collect_vars	Off	Off
<b>xdebug</b> .coverage_enable	On	On
<b>xdebug</b> .default_enable	On	On

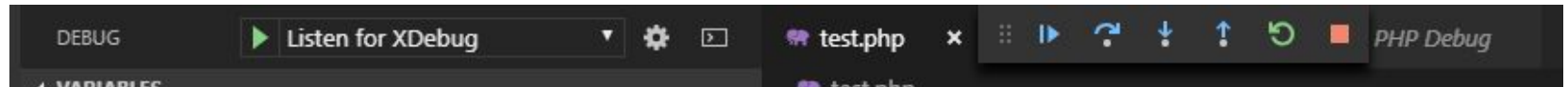


# Uso de XDebug

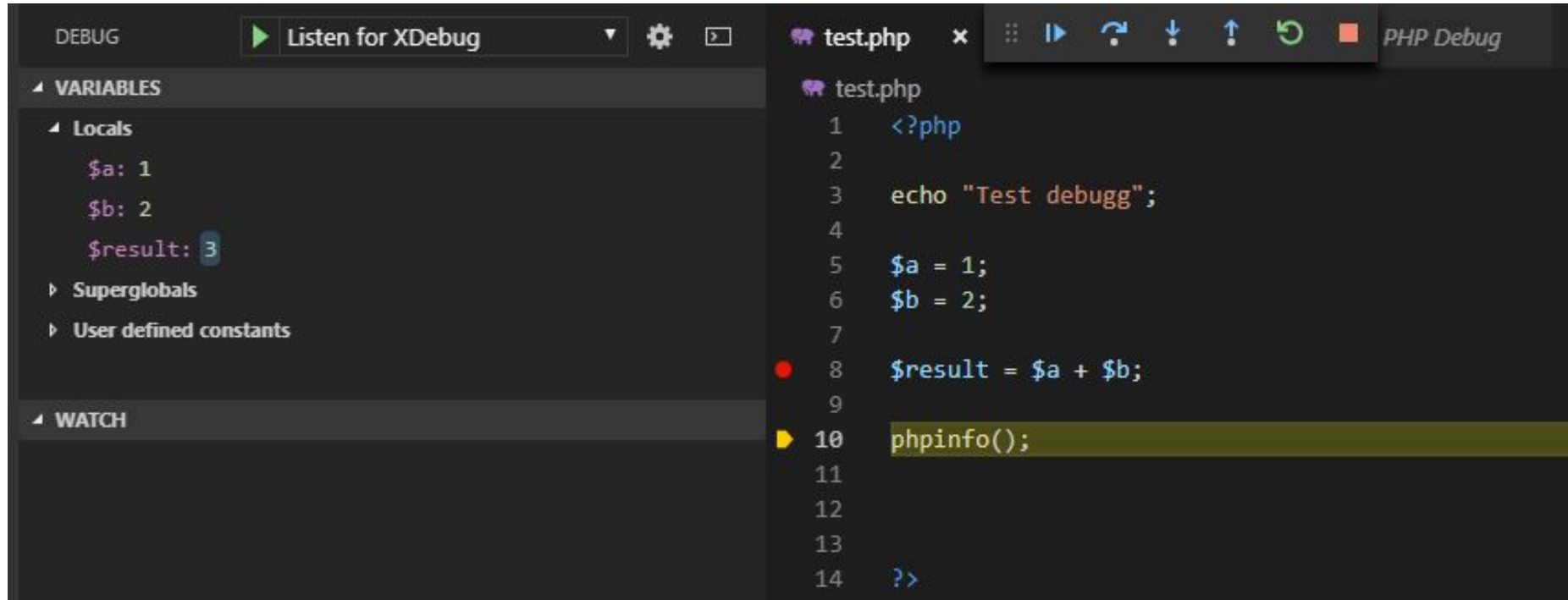
1. Es importante hacer click en “Add Configuration...” y seleccionar XDebug dentro de la sección de Debug



2. Hacer click en Start Debug



# Uso de XDebug



The image shows a screenshot of an IDE interface with a dark theme, displaying the XDebug debugging environment. The interface is divided into three main sections: a top toolbar, a left sidebar for variables, and a main editor for the code.

**Top Toolbar:** Features a "DEBUG" button, a "Listen for XDebug" button with a green play icon, a settings gear icon, and a "PHP Debug" button with a red square icon. To the right of these are several navigation icons: a list icon, a play icon, a refresh icon, a download icon, an upload icon, a circular arrow icon, and a red square icon.

**Left Sidebar (VARIABLES):** Contains a tree view with the following sections:

- VARIABLES** (expanded)
  - Locals** (expanded)
    - `$a: 1`
    - `$b: 2`
    - `$result: 3` (highlighted with a blue selection box)
  - Superglobals** (collapsed)
  - User defined constants** (collapsed)
- WATCH** (collapsed)

**Main Editor:** Displays the code for `test.php`. The code is as follows:

```
1 <?php
2
3 echo "Test debugg";
4
5 $a = 1;
6 $b = 2;
7
8 $result = $a + $b;
9
10 phpinfo();
11
12
13
14 ?>
```

The code is syntax-highlighted. Line 10, `phpinfo();`, is highlighted with a green background. A red dot is visible next to line 8, indicating a breakpoint. The editor title bar shows "test.php" and a close button.