# C1 - Tổng quan về ngôn ngữ lập trình PHP

## 1. Giới thiệu PHP

**PHP** (Hypertext Preprocessor) là ngôn ngữ lập trình được dùng cho mục đích tổng quát, ban đầu nó được thiết kể tập trung cho các ứng dụng web. Được giới thiệu từ năm 1994 bởi Rasmus Lerdorf.

PHP có tốc độ nhanh, nhỏ gọn, cú pháp giống C và Java, dễ học và thời gian xây dựng sản phẩm tương đối ngắn hơn so với các ngôn ngữ khác nên PHP đã nhanh chóng trở thành một ngôn ngữ lập trình web phổ biến nhất thế giới.

### **1.1. Lịch sử phát triển**

Phiên bản PHP hiện tại là PHP 7.x, phiên bản PHP mới này có ứng dụng rất nhiều kỹ thuật lập trình hiện đại như lớp ẩn danh, chỉ rõ kiểu trả về cho hàm ... Các cập nhật mới về PHP tại <https://www.php.net/>

#### **PHP/FI (1994)**

PHP được phát triển từ một sản phẩm có tên là PHP/FI. PHP/FI do Rasmus Lerdorf tạo ra năm 1994, ban đầu được xem như là một tập con đơn giản của các mã kịch bản Perl để theo dõi tình hình truy cập đến bản sơ yếu lý lịch của ông trên mạng. Ông đã đặt tên cho bộ mã kịch bản này là 'Personal Home Page Tools'. Khi cần đến các chức năng rộng hơn, Rasmus đã viết ra một bộ thực thi bằng C lớn hơn để có thể truy vấn tới các cơ sở dữ liệu và giúp cho người sử dụng phát triển các ứng dụng web đơn giản. Rasmus đã quyết định công bố mã nguồn của PHP/FI cho mọi người xem, sử dụng cũng như sửa các lỗi có trong nó đồng thời cải tiến mã nguồn.

Vào năm 1997, PHP/FI 2.0, lần viết lại thứ hai của phiên bản C, đã thu hút được hàng ngàn người sử dụng trên toàn thế giới với xấp xỉ 50.000 tên miền đã được ghi nhận là có cài đặt nó, chiếm khoảng 1% số tên miền có trên mạng Internet. Tuy đã có tới hàng nghìn người tham gia đóng góp vào việc tu chỉnh mã nguồn của dự án này thì vào thời đó nó vẫn chủ yếu chỉ là dự án của một người.

#### **PHP 3 (1998)**

PHP 3.0 là phiên bản đầu tiên cho chúng ta thấy một hình ảnh gần gũi với các phiên bản PHP mà chúng ta được biết ngày nay. Nó đã được Andi Gutmans và Zeev Suraski tạo ra năm 1997 sau khi viết lại hoàn toàn bộ mã nguồn trước đó. Lý do chính mà họ đã tạo ra phiên bản này là do họ nhận thấy PHP/FI 2.0 hết sức yếu kém trong việc phát triển các ứng dụng thương mại điện tử mà họ đang xúc tiến trong một dự án của trường đại học. Trong một nỗ lực hợp tác và bắt đầu xây dựng dựa trên cơ sở người dùng đã có của PHP/FI, Andi, Rasmus và Zeev đã quyết định hợp tác và công bố PHP 3.0 như là phiên bản thế hệ kế tiếp của PHP/FI 2.0, và chấm dứt phát triển PHP/FI 2.0.

Một trong những sức mạnh lớn nhất của PHP 3.0 là các tính năng mở rộng mạnh mẽ của nó. Ngoài khả năng cung cấp cho người dùng cuối một cơ sở hạ tầng chặt chẽ dùng cho nhiều cơ sở dữ liệu, giao thức và API khác nhau, các tính năng mở rộng của PHP 3.0 đã thu hút rất nhiều nhà phát triển tham gia và đề xuất các mô đun mở rộng mới. Hoàn toàn có thể kết luận được rằng đây chính là điểm mấu chốt dẫn đến thành công vang dội của PHP 3.0. Các tính năng khác được giới thiệu trong PHP 3.0 gồm có hỗ trợ cú pháp hướng đối tượng và nhiều cú pháp ngôn ngữ nhất quán khác.

Ngôn ngữ hoàn toàn mới đã được công bố dưới một cái tên mới, xóa bỏ mối liên hệ với việc sử dụng vào mục đích cá nhân hạn hẹp mà cái tên PHP/FI 2.0 gợi nhắc. Nó đã được đặt tên ngắn gọn là 'PHP', một kiểu viết tắt hồi quy của "PHP: Hypertext Preprocessor".

Vào cuối năm 1998, PHP đã phát triển được con số cài đặt lên tới hàng chục ngàn người sử dụng và hàng chục ngàn Web site báo cáo là đã cài nó. Vào thời kì đỉnh cao, PHP 3.0 đã được cài đặt cho xấp xỉ 10% số máy chủ Web có trên mạng Internet.

PHP 3.0 đã chính thức được công bố vào tháng 6 năm 1998, sau thời gian 9 tháng được cộng đồng kiểm nghiệm.

#### **PHP 4 (2000)**

Vào mùa đông năm 1998, ngay sau khi PHP 3.0 chính thức được công bố, Andi Gutmans và Zeev Suraski đã bắt đầu bắt tay vào việc viết lại phần lõi của PHP. Mục đích thiết kế là nhằm cải tiến tốc độ xử lý các ứng dụng phức tạp, và cải tiến tính mô đun của cơ sở mã PHP. Những ứng dụng như vậy đã chạy được trên PHP 3.0 dựa trên các tính năng mới và sự hỗ trợ khá nhiều các cơ sở dữ liệu và API của bên thứ ba, nhưng PHP 3.0 đã không được thiết kế để xử lý các ứng dụng phức tạp như thế này một cách có hiệu quả.

Một động cơ mới, có tên 'Zend Engine' (ghép từ các chữ đầu trong tên của Zeev và Andi), đã đáp ứng được các nhu cầu thiết kế này một cách thành công, và lần đầu tiên được giới thiệu vào giữa năm 1999. PHP 4.0, dựa trên động cơ này, và đi kèm với hàng loạt các tính năng mới bổ sung, đã chính thức được công bố vào tháng 5 năm 2000, gần 2 năm sau khi bản PHP 3.0 ra đời. Ngoài tốc độ xử lý được cải thiện rất nhiều, PHP 4.0 đem đến các tính năng chủ yếu khác gồm có sự hỗ trợ nhiều máy chủ Web hơn, hỗ trợ phiên làm việc HTTP, tạo bộ đệm thông tin đầu ra, nhiều cách xử lý thông tin người sử dụng nhập vào bảo mật hơn và cung cấp một vài các cấu trúc ngôn ngữ mới.

Với PHP 4, số nhà phát triển dùng PHP đã lên đến hàng trăm nghìn và hàng triệu site đã công bố cài đặt PHP, chiếm khoảng 20% số tên miền trên mạng Internet.

Nhóm phát triển PHP cũng đã lên tới con số hàng nghìn người và nhiều nghìn người khác tham gia vào các dự án có liên quan đến PHP như PEAR, PECL và tài liệu kĩ thuật cho PHP.

#### **PHP 5 (2005)**

Sự thành công hết sức to lớn của PHP 4.0 đã không làm cho nhóm phát triển PHP tự mãn. Cộng đồng PHP đã nhanh chóng giúp họ nhận ra những yếu kém của PHP 4 đặc biệt với khả năng hỗ trợ lập trình hướng đối tượng (OOP), xử lý XML, không hỗ trợ giao thức máy khách mới của MySQL 4.1 và 5.0, hỗ trợ dịch vụ web yếu. Những điểm này chính là mục đích để Zeev và Andi viết Zend Engine 2.0, lõi của PHP 5.0. Một thảo luận trên Slashdot đã cho thấy việc phát triển PHP 5.0 có thể đã bắt đầu vào thời điểm tháng 12 năm 2002 nhưng những bài phỏng vấn Zeev liên quan đến phiên bản này thì đã có mặt trên mạng Internet vào khoảng tháng 7 năm 2002. Ngày 29 tháng 6 năm 2003, PHP 5 Beta 1 đã chính thức được công bố để cộng đồng kiểm nghiệm. Đó cũng là phiên bản đầu tiên của Zend Engine 2.0. Phiên bản Beta 2 sau đó đã ra mắt vào tháng 10 năm 2003 với sự xuất hiện của hai tính năng rất được chờ đợi: Iterators, Reflection nhưng namespaces một tính năng gây tranh cãi khác đã bị loại khỏi mã nguồn. Ngày 21 tháng 12 năm 2003: PHP 5 Beta 3 đã được công bố để kiểm tra với việc phân phối kèm với Tidy, bỏ hỗ trợ Windows 95, khả năng gọi các hàm PHP bên trong XSLT, sửa chữa nhiều lỗi và thêm khá nhiều hàm mới. PHP năm bản chính thức đã ra mắt ngày 13 tháng 7 năm 2004 sau một chuỗi khá dài các bản kiểm tra thử bao gồm Beta 4, RC 1, RC2, RC3. Mặc dù coi đây là phiên bản sản xuất đầu tiên nhưng PHP 5.0 vẫn còn một số lỗi trong đó đáng kể là lỗi xác thực HTTP.

Ngày 14 tháng 7 năm 2005, PHP 5.1 Beta 3 được PHP Team công bố đánh dấu sự chín muồi mới của PHP với sự có mặt của PDO, một nỗ lực trong việc tạo ra một hệ thống API nhất quán trong việc truy cập cơ sở dữ liệu và thực hiện các câu truy vấn. Ngoài ra, trong PHP 5.1, các nhà phát triển PHP tiếp tục có những cải tiến trong nhân Zend Engine 2, nâng cấp mô đun PCRE lên bản PCRE 5.0 cùng những tính năng và cải tiến mới trong SOAP, streams và SPL.

#### **PHP 6**

Phiên bản không được phát hành

#### **PHP 7 (2015)**

Cải tiến tốc độ xử lý của PHP và hỗ trợ các tính năng mới như: khai báo kiểu dữ liệu chính xác, toán tử kiểm tra null ??, lớp ẩn danh, từ khóa use để sử dụng namespace ...

### **1.2 Vai trò của PHP**

PHP là ngôn ngữ phổ biến trên thế giới được sử dụng bởi nhiều công ty / doanh nghiệp lớn như Google, Facebok, WordPress, Wikipedia, Yahoo ...

Hiện nay số lượng ứng dụng web chạy bằng PHP trên thế giới khoảng 80%

PHP dễ học, kế thừa cú pháp từ ngôn ngữ C, PHP có thể tạo ra các loại ứng dụng đa nền tảng:

* Ứng dụng chạy trên máy chủ (Server-side script)
* Các ứng dụng Desktop có UI ([GUI Extensions](https://php.net/ui))
* Các ứng dụng chạy với dòng lệnh

Do gần với C / C++, hỗ trợ lập trình hướng đối tượng nên có thể dùng PHP để học lập trình cơ bản, từ nền tảng đó có thể học các loại ngôn ngữ khác như Java, C# ... dễ dàng hơn.

## 2. Cài đặt môi trường phát triển ứng dụng PHP

Để phát triển ứng dụng PHP cần chuẩn bị môi trường phát triển (gồm phần mềm PHP), nếu cài đặt đầy đủ một hệ thống thì cần có thêm máy chủ web như (Apache, IIS ...) và hệ quản trị cơ sở dữ liệu (như MySQL, SQL Server, MariaDb ...). Ở đây sẽ chọn cài đặt **WampServer** vì nó bao gồm PHP, Apache, MySQL

Chọn một IDE (Môi trường phát triển tích hợp - Integrated Development Environment) thích hợp. Có rất nhiều IDE để chọn như [PHP Storm](https://www.jetbrains.com/phpstorm/eap/), [Netbeans](https://netbeans.org/), [ZendStudio](https://www.zend.com/en/products/studio), [Visual Studio](https://visualstudio.microsoft.com/), [Visual Studio Code](https://code.visualstudio.com/) ... Ở đây chọn sử dụng **Visual Studio Code**

### **2.1. Cài đặt PHP, APACHE, MYSQL với WampServer**

**WampServer** là một phần mềm phối hợp nền tảng dành cho Windows, nó bao gồm máy chủ Apache, ngôn ngữ lập trình PHP và hệ quản trị CSDL MySQL. Với WampServer nó cung cấp môi trường hoàn chỉnh để chạy PHP

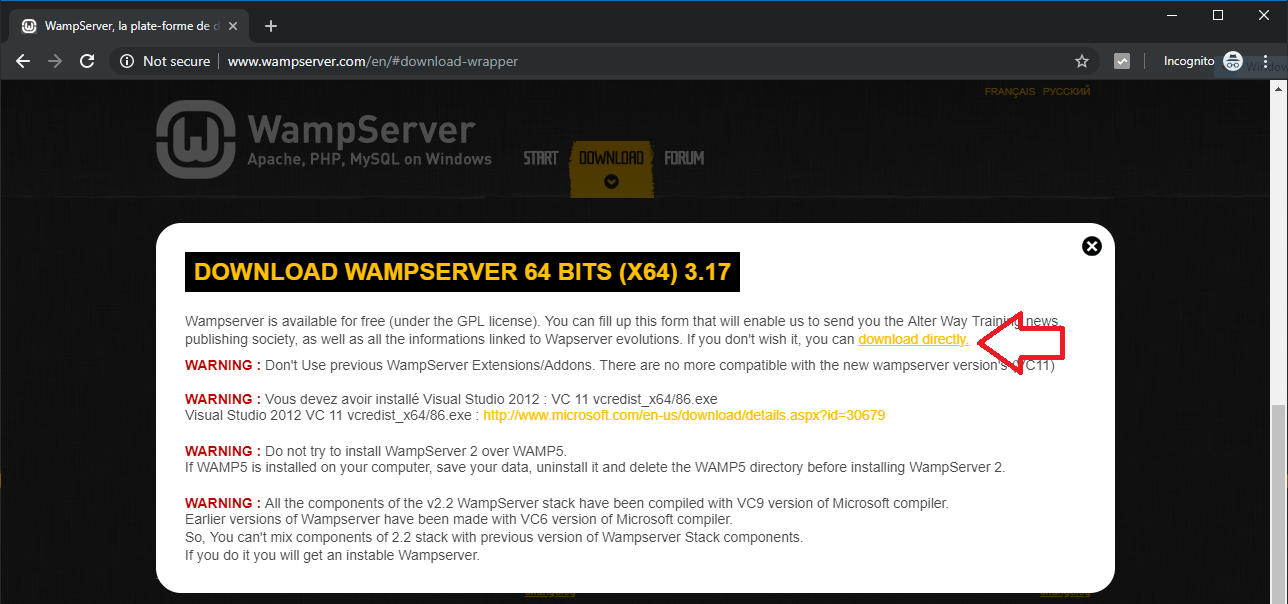
**B1) Cài đặt thư viện Visual C++**

Đảm bảo Windows có cài đặt Visual C++ Redistributable for Visual Studio 2012 <https://www.microsoft.com/en-us/download/details.aspx?id=30679> (vào đường link và chọn đúng phiên bản phù hợp với hệ điều hành để tải về).

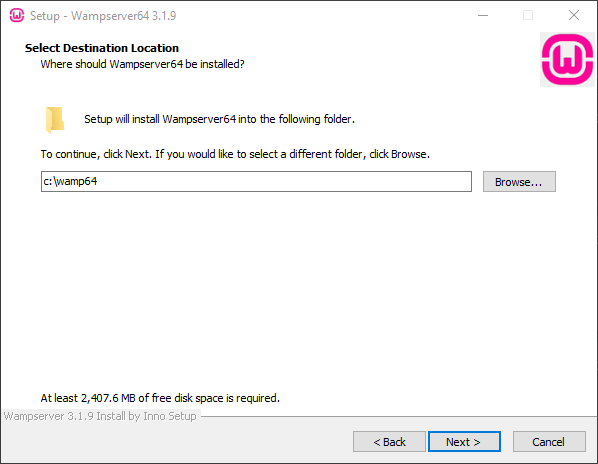
**B2) Cài đặt WampServer**

Tải và cài đặt WampServer tại địa chỉ: [http://www.wampserver.com/en/](http://www.wampserver.com/en/#download-wrapper)

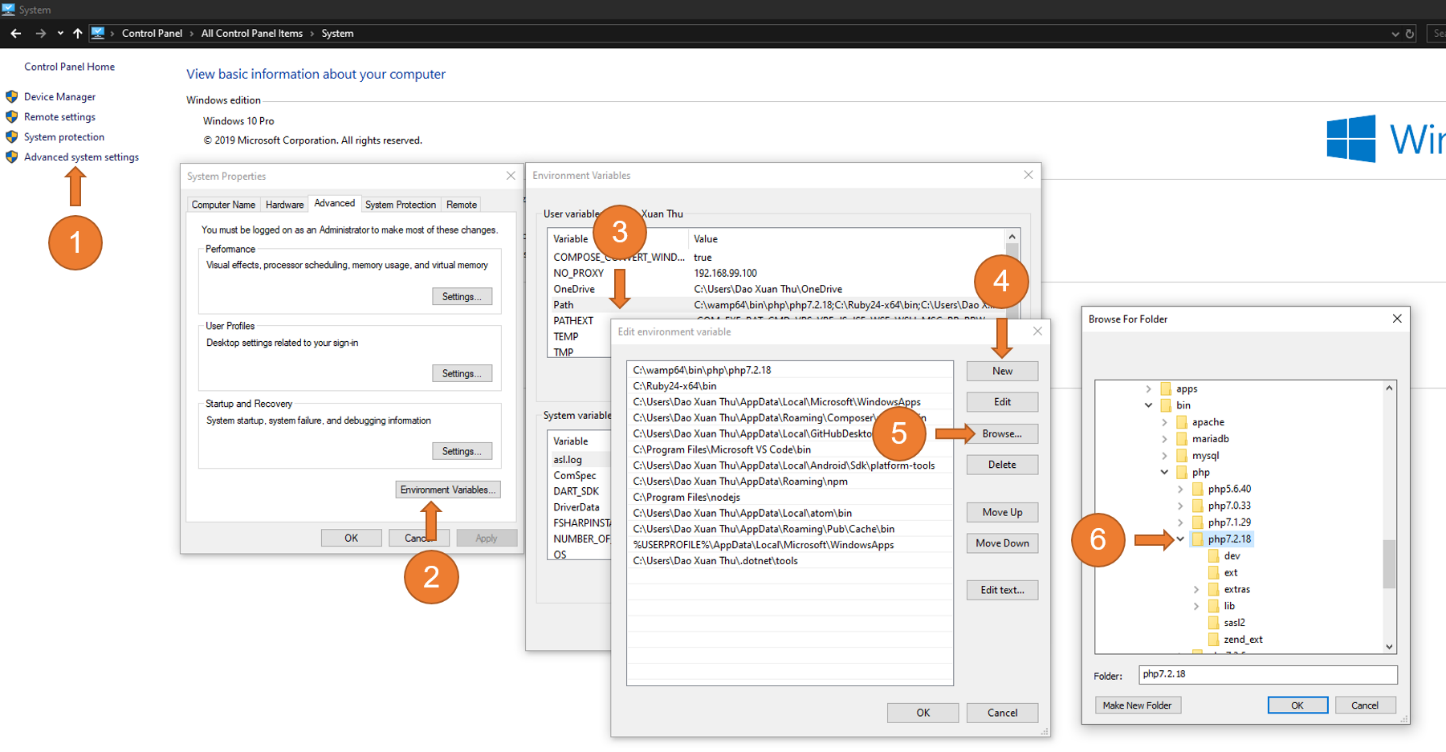
Chọn đúng phiên bản và bấm vào download directly để tải về.



Tiến hành chạy cài đặt chọn thư mục cài đặt là c:\wamp64



**B3) Thiết lập biến môi trường PATH**



Mở **Contol Panel > System** rồi chọn **1** **Advanced system settings**

**2** chọn mục **Environment Variables** của hộp thoại **System Properties**

Tại hộp thoại **Environment Variables** kích đúp vào **3** **Path**

Tại hộp thoại **Edit Enviroment Variable** bấm vào mục **4** **New ...** để tạo ra một biến môi trường mới. Rồi bấm vào **5** **Browse..** để gán đường dẫn PHP cho biến môi trườn này

Ở cửa sổ **Browse for Folder** chọn đến đường dẫn cài đặt **6** PHP đó là: C:\wamp64\bin\php\php7.2.18

Nhấn OK để đóng các hộp thoại

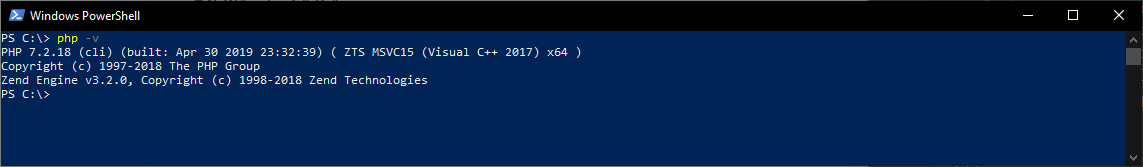
**B4) Kiểm tra hệ thống**

Sau khi hoàn thành cài đặt và thiết lập biến môi trường như trên hãy khởi động lại máy.

**Kiểm tra PHP** mở cmd hoặc PowerShell và gõ lệnh:

php -v

Nếu có xuất hiện thông tin phiên bản PHP thì php cài đặt thành công

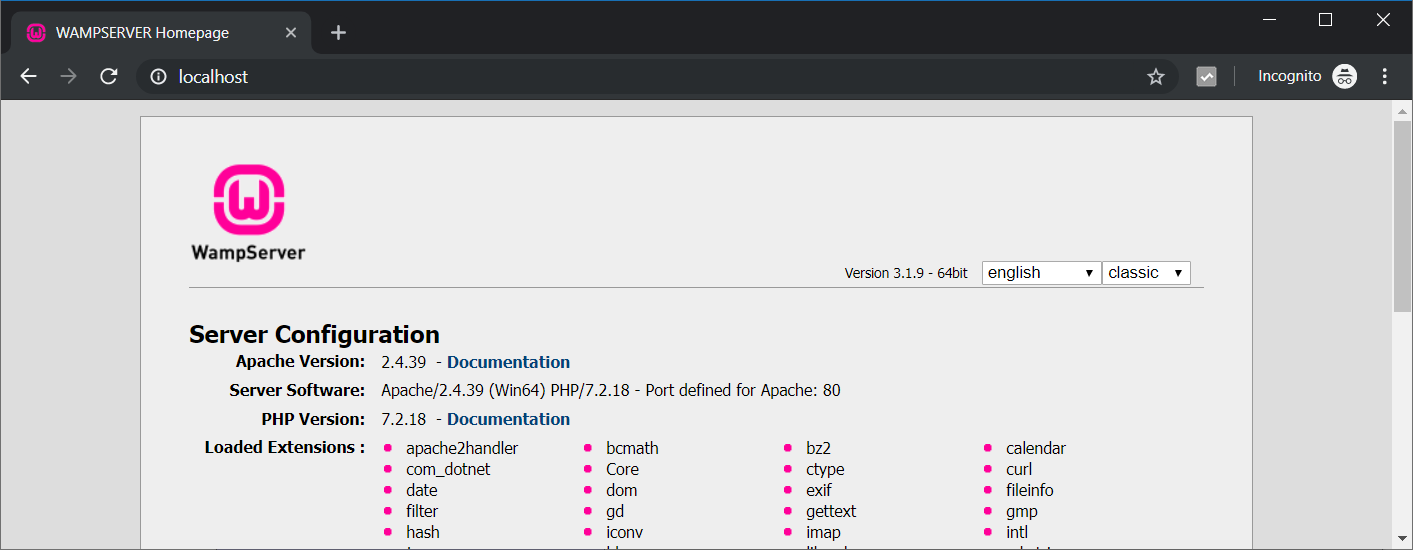


**Kiểm tra WampServer** (chạy 3 dịch vụ Apache, MySQL, PHP)

Khởi động WampServer bằng cách kích đúp biểu tượng của nó trên Desktop. Sau khi WampServer khởi động xong có biểu tượng icon màu xanh trên TaskBar chứng tỏ tất các các dịch vụ cần thiết đang chạy.

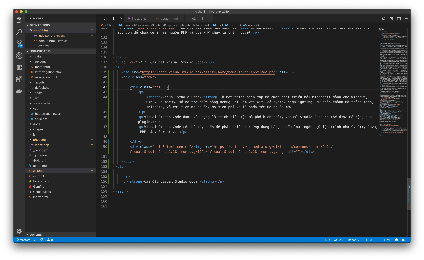
Mặc định máy chủ Apache lắng nghe ở cổng 80, có thể truy cập trừ trình duyệt với địa chỉ http://127.0.0.1 hoặc http://localhost

Nếu từ trình duyệt thấy nội dung như sau thì WampServer đã hoạt động chính xác:



Thư mục làm việc mặc định là C:\wamp64\www, trong thư mục này có thể tạo các thư mục con để chứa dự án mã nguồn PHP và gọi PHP chạy từ trình duyệt

### **2.2. Cài đặt Visual Studio Code**

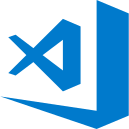


Màn hình Visual Studio Code

Visual Studio Code là một trình biên tập mã được phát triển bởi Microsoft dành cho Windows, Linux và macOS. Nó hỗ trợ chức năng debug, đi kèm với Git, có syntax highlighting, tự hoàn thành mã thông minh, snippets, và cải tiến mã nguồn. Nó miễn phí và là phần mềm mã nguồn mở.

Visual Studio Code được xếp hạng là trình biên tập mã phổ biến nhất, Visual Studio Code có thể được mở rộng qua plugin.

Visual Studio Code có thể tùy biến để phát triển các ứng dụng bằng nhiều loại ngôn ngữ lập trình như C, C++, Java, PHP, JavaScript ...



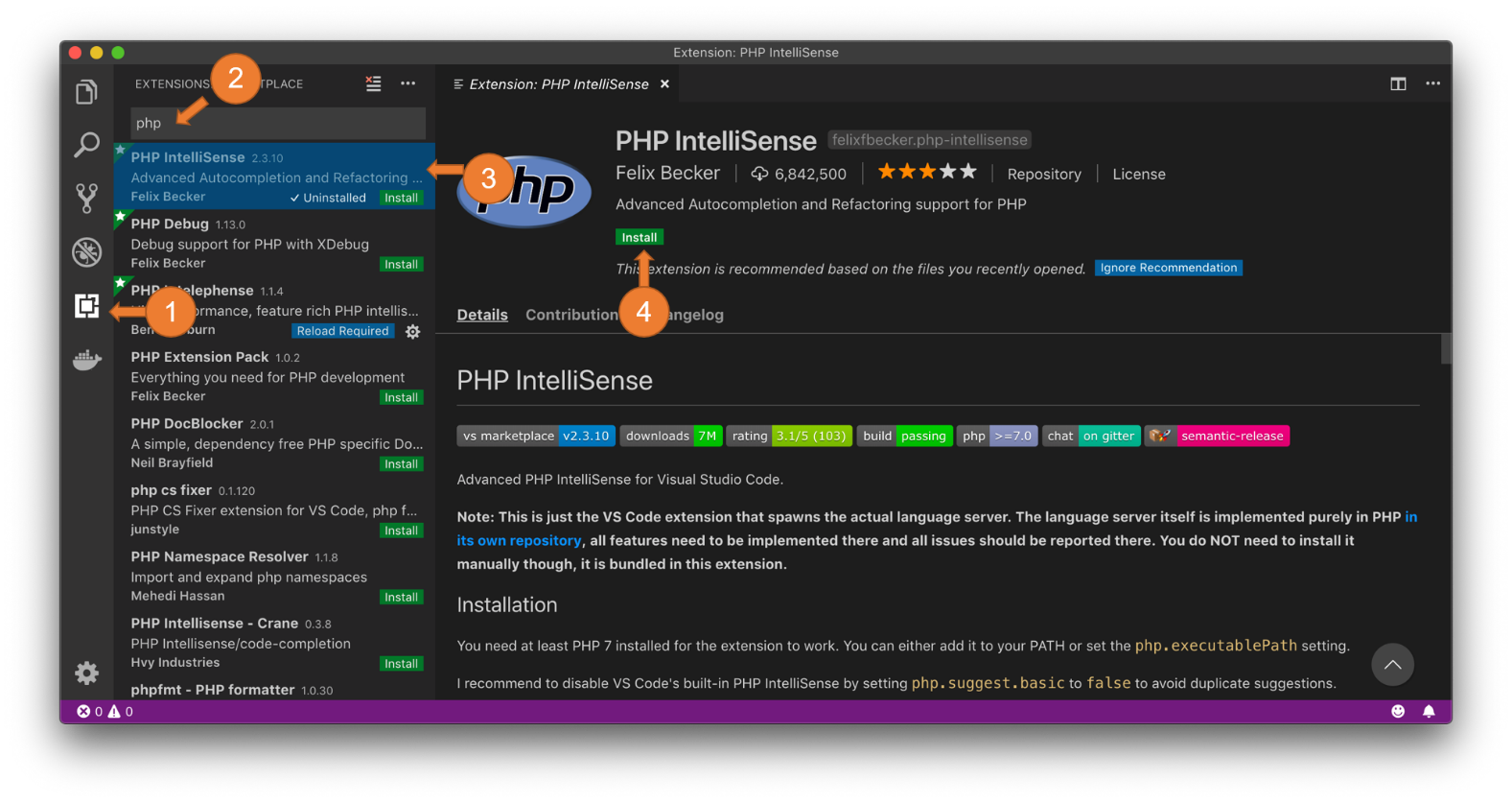
**Cài đặt Visual Studio Code**

Tải về và tiến hành chạy file cài đặt tại <https://code.visualstudio.com/>, sau khi hoàn thành cài đặt chạy chương trình nó có giao diện như hình trên.

**Cài đặt phần mở rộng để phát triển PHP**

**PHP IntelliSense**

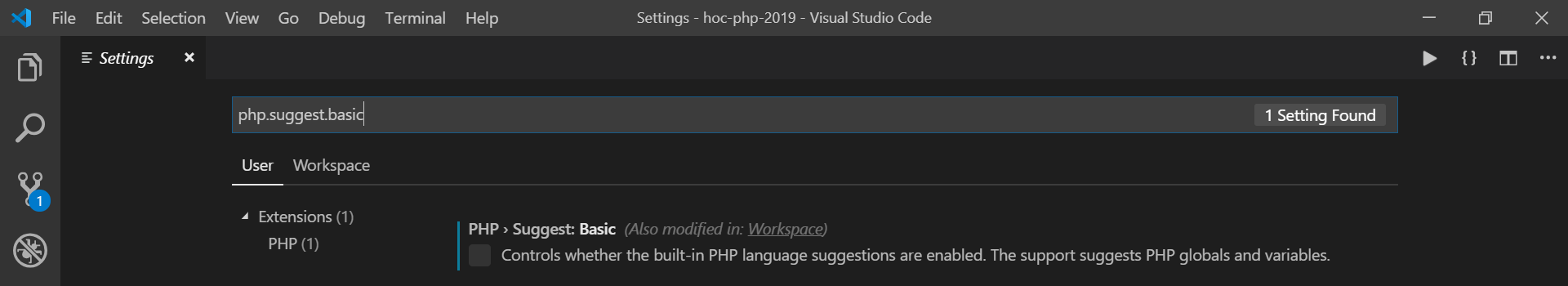
Extension này hỗ trợ tự động hoàn thành câu lệnh PHP khi đang viết, tự động xuất hiện các gợi ý, hỗ trợ tham khảo về cú pháp PHP ...



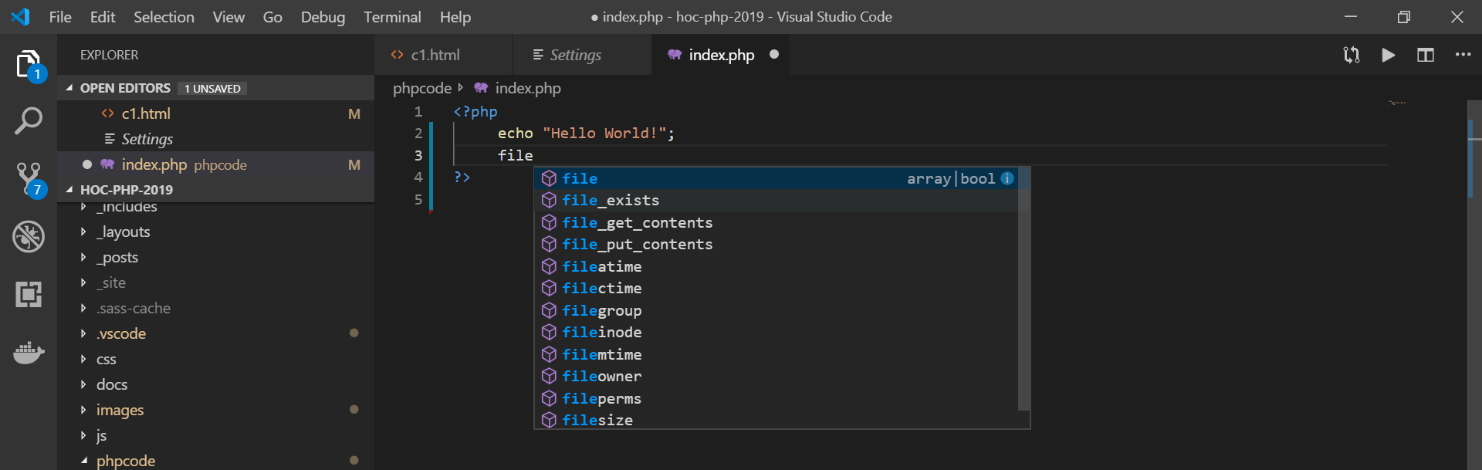
Chạy VSC, chọn biểu tượng **Extensions** **1**, sau đó gõ vào mục tìm kiếm từ khóa php **2**

Chọn Extension có tên **PHP IntelliSense** **3** sau đó bấm vào mục **Install** **4**

Để tránh xung đột cần thiết lập php.suggest.basic bằng false. Mở thiết lập VSC **Setting**bằng phím CTRL,, sau đó gõ vào php.suggest.basic, bỏ đánh dấu mục Controls whether the build-in PHP language ...



Khi soạn code cần gợi ý nhấn phím ctrl + space

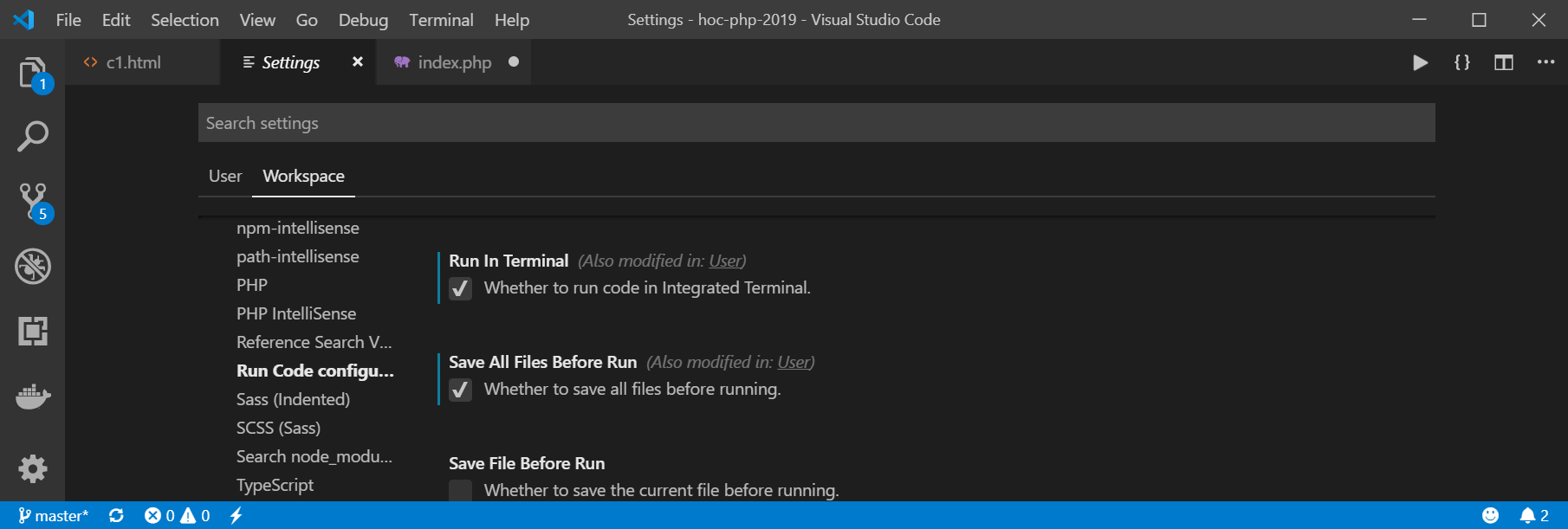


Bằng cách cài Extension tương tự, tiến hành cài đặt thêm các Extension sau:

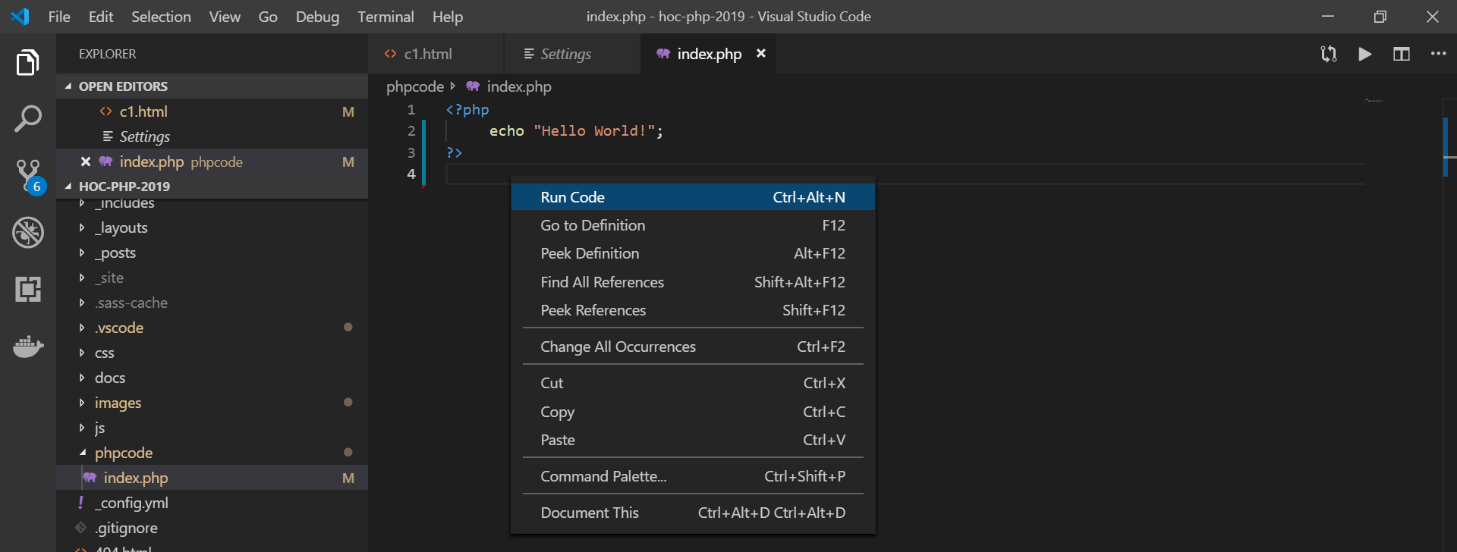
**Code Runner**

Cho phép chạy các ngôn ngữ lập trình: C, C++, Java, JavaScript, PHP, Python, Perl, Perl 6, Ruby, Go, Lua, Groovy, PowerShell, BAT/CMD, BASH/SH, F# Script, F# (.NET Core), C# Script, C# (.NET Core), VBScript, TypeScript, CoffeeScript, Scala, Swift, Julia, Crystal, OCaml Script, R, AppleScript, Elixir, Visual Basic .NET, Clojure, Haxe, Objective-C, Rust, Racket, AutoHotkey, AutoIt, Kotlin, Dart, Free Pascal, Haskell, Nim, D, Lisp, Kit, ...

Mở **Setting**, chọn mục **Run Code Configution** sau đó đánh dấu vào các mục: Run In Terminal, Save All Files Before Run



Có thêm menu context (nhấn phải chuột vào tên file) **Run Code**, chọn nó sẽ thi hành code ngay lập tức



**HTML CSS Support**

Hỗ trợ biên tập mã HTML, CSS

**IntelliSense for CSS class names in HTML**

Hỗ trợ gợi ý, tự động điền code khi gán class CSS vào HTML

### **2.3. Tạo mới, ghi một chương trình PHP**

**2.3.1. Chạy mã PHP từ dòng lệnh**

Khi bắt đầu một dự án PHP, nên tạo một thư mục dự án, trong thư mục đó dành để chứa các file, thư mục mã nguồn, dữ liệu *(Hình ảnh, Video, CSS, JS ...)* của dự án.

Các File mã nguồn PHP có phần mở động là .php

Ví dụ tạo ra một thư mục dự án tên là app1, trong thư mục này tạo ra một file mã nguồn PHP đầu tiên đặt tên là vidu1.php

Có thể dùng PowerShell vào thư mục dự án app1 bằng lệnh cd để tạo file bằng lệnh:

echo $null >> vidu1.php

Hoặc dùng trực tiếp Visual Studio Code để tạo và mở soạn thảo với lệnh code vidu1.php

Dùng chương trình soạn thảo text bất kỳ (Notepad, NotePad++, Visual Studio Code ...) để soạn thảo với nội dung sau:

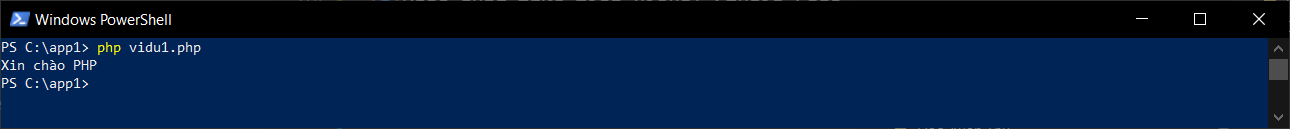
<?php

echo "Xin chào PHP";

?>

Chạy đoạn mã PHP này sẽ in ra dòng Xin chào PHP

Để chạy vidu1.php dùng PS vào thư mục chứa dự án và gõ lệnh: php vidu1.php

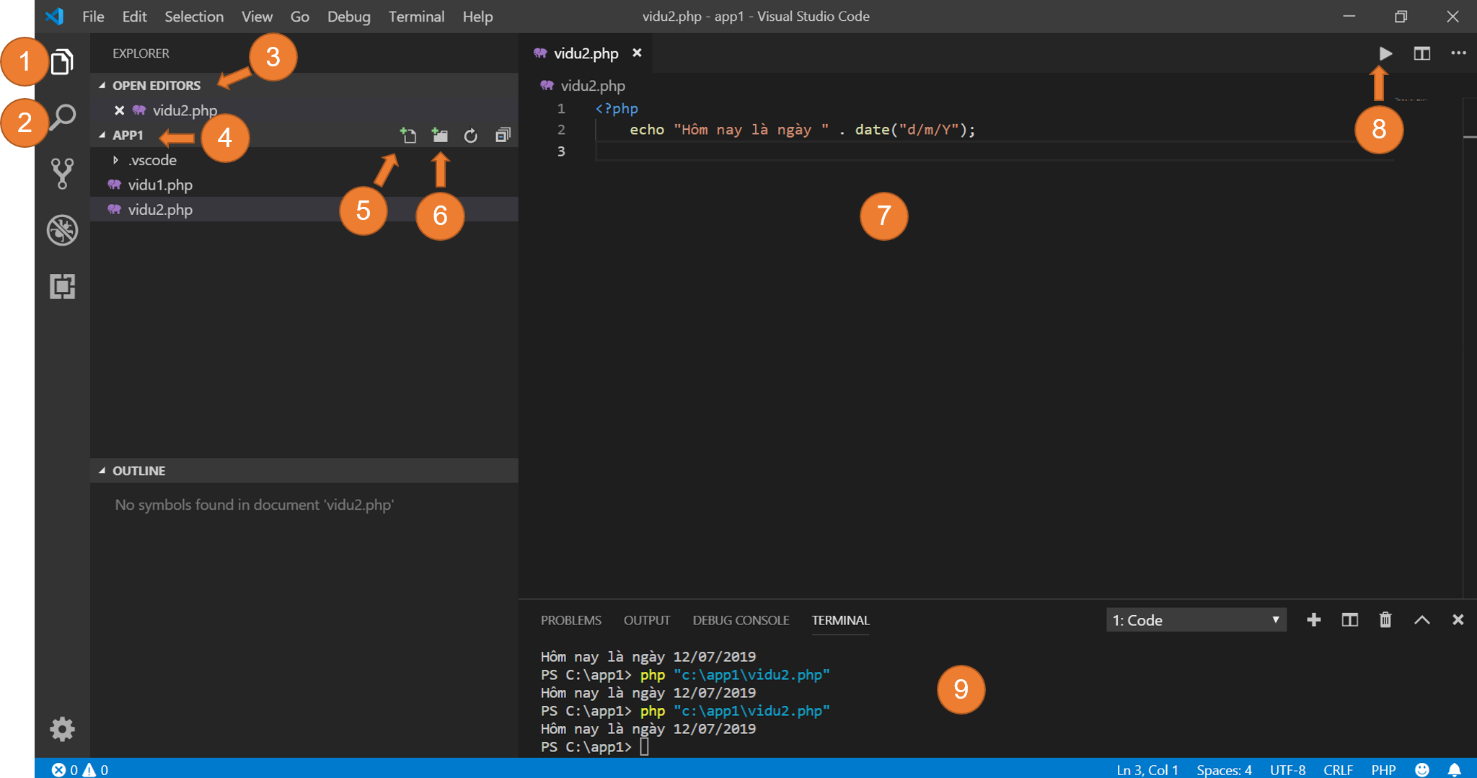


Như vậy đã chạy một đoạn mã PHP thành công

**2.3.2. Soạn thảo mã PHP, chạy PHP từ VSC**

Chạy VSC, mở thư mục chứa dự án bằng menu lệnh: File > Open Folder ...

Nếu đang từ PowerShell gõ lệnh code thu\_muc để mở thu\_muc, hoặc code . để mở thư mục hiện tại



* **1** Bật duyệt file trên thư mục dự án
* **2** Tìm kiếm file (theo nội dung chứa trong file)
* **3** Khu vực hiệ thị các file đang được mở để soạn thảo
* **4** Thư mục dự án và các file, folder trong dự án. Để mở một file nháy kép vào tên file.
* **5** Tạo file mới
* **6** Tạo thư mục mới
* **7** Khu vực soạn thảo file đang mở
* **8** Thi hành file (chạy mã - code)
* **9** Terminal (CMD) tích hợp vào VSC

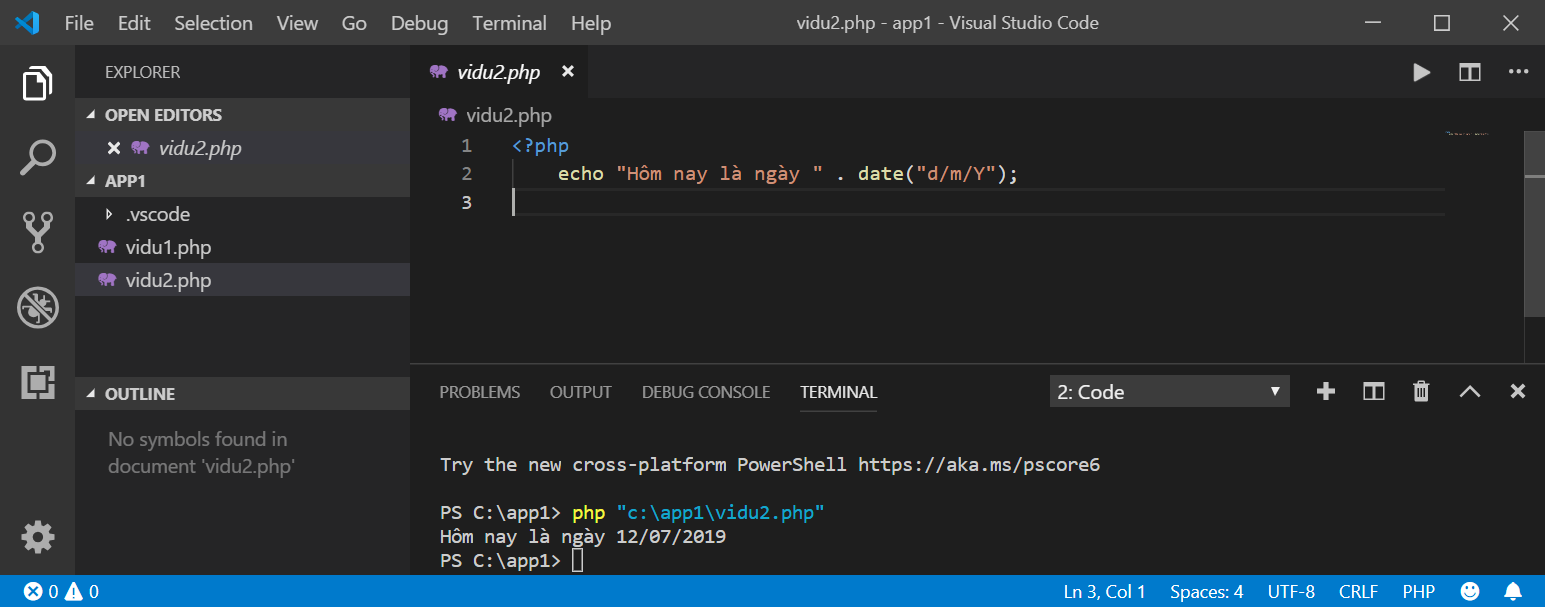
Ví dụ bấm vào **9** tạo file mới có tên là vidu2.php

Sau đó nhập nội dung:

<?php

echo "Hôm nay là ngày " . date("d/m/Y");

Bấm vào chạy code **8**



**2.3.3. Chạy thư mục code PHP như một trang web**

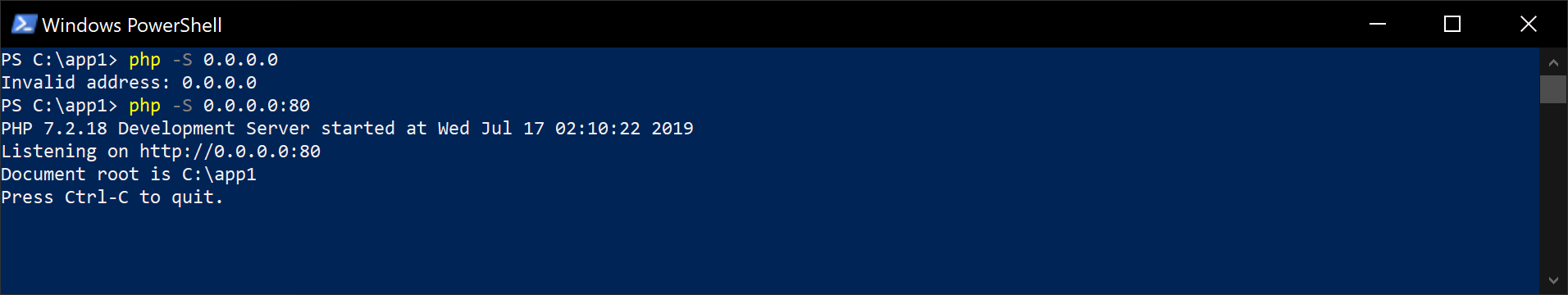
Để chạy một thư mục code PHP bằng cách truy cập chúng từ trình duyệt (chạy một Webserver PHP) ở đây sẽ tạo máy chủ Webserver trực tiếp từ PHP (chưa dùng đến Apache Webserver) thực hiện như sau:

Từ Terminal vào thư mục chứa code PHP, gõ lệnh:

php -S 0.0.0.0:8080

Trong đó cổng 8080 tự chọn để tránh xung đột với cổng 80 mặc định. Nếu muốn sử dụng cổng 80 hãy dừng Apache và gõ lệnh:

php -S 0.0.0.0:80



Khi Webserver do PHP tạo ra đang chạy thì có thể truy cập đến ứng dụng thông qua trình duyệt, ở các địa chỉ như http://localhost:8080, http://127.0.0.1:8080 hoặc từ máy khác bằng http://ip:8080

Mặc định nó sẽ tìm và chạy file index.php, nếu muốn chạy các file khác gõ theo địa chỉ http://localhost:8080/vidu1.php

Kết thúc Webserver bằng cách gõ vào Terminal phím CTRL + C

**2.3.4. Thẻ php**

Các đoạn mã PHP được đặt trong thẻ php mở đầu là <php và kết thúc bởi ký hiệu >?

<?php

// Mã PHP ở đây

?>

Nội dung ngoài thẻ php, thường là các mã HTML

<?php

//...

?>

Những nội dung văn bản bên ngoài thẻ php mặc định khi chạy nó xuất ra đúng nội dung đó.

Ví dụ, HTML và PHP pha trộn với nhau: (vidu3.php)

<!DOCTYPE html>

<html lang="en">

<head>

<meta charset="UTF-8">

<meta name="viewport" content="width=device-width, initial-scale=1.0">

<meta http-equiv="X-UA-Compatible" content="ie=edge">

<title>Đây là trang HTML có mã PHP</title>

</head>

<body>

<h1>Ví dụ PHP</h1>

<?php

$a = 10;

$b = 2.2;

$c = $a \* $b;

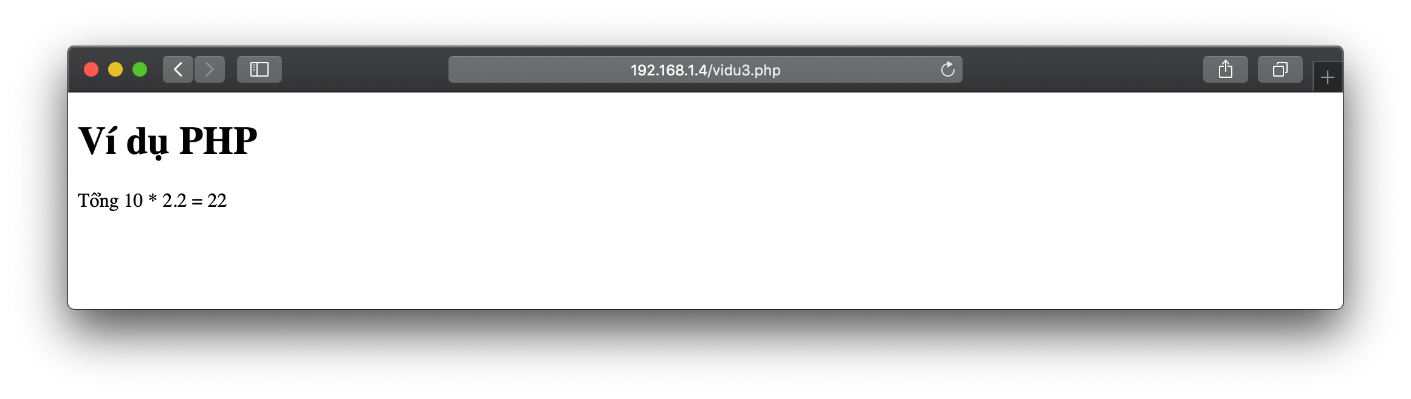
echo "Tổng $a \* $b = $c";

?>

</body>

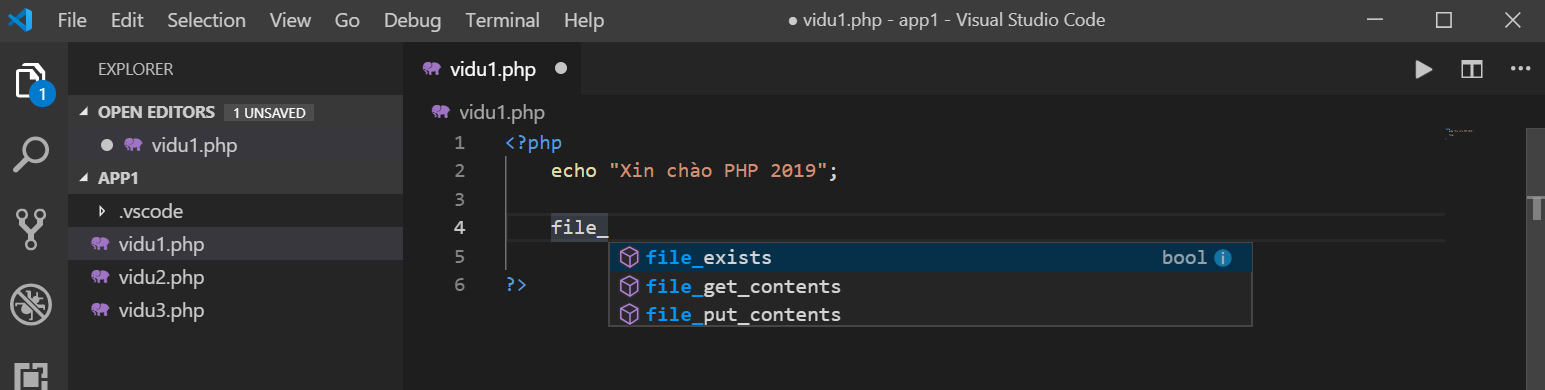
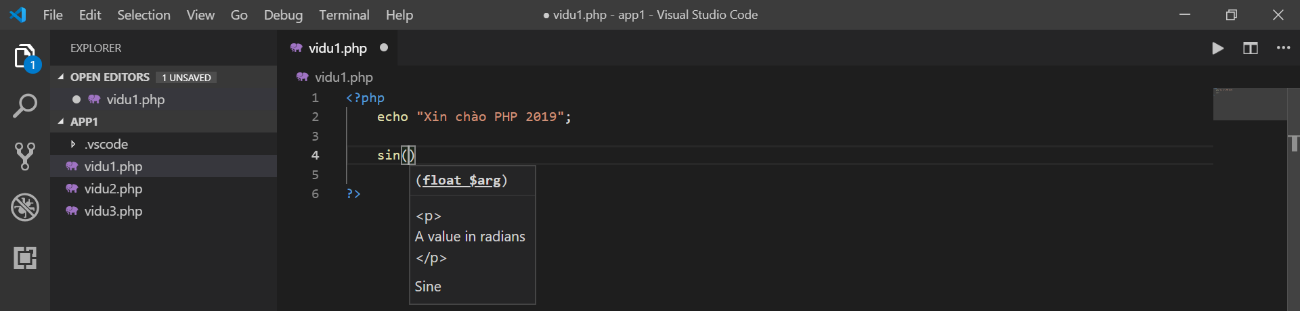
</html>

Thực hiện chạy file trên qua trình duyệt



## 3. Sử dụng sự trợ giúp

Trong quá trình phát triển dự án PHP, các nguồn tài nguyên sau có thể tham khảo về cú pháp và các thư viện

* Tham khảo về cú pháp và các đặc điểm ngôn ngữ lập trình PHP từ website của PHP: <https://www.php.net/manual/en/langref.php>
* Khi đang soạn thảo code, gõ một phần tên hàm - cần sự gợi ý từ các Extension đã cài đặt gõ tổ hợp phím CTRL + SPACEBAR ở cuối từ cần gõ (xem hình dưới).   
  
* Khi tham khảo về tham số của một hàm, để con trỏ ở phần tham số hàm và nhấn tổ hợp phím CTRL+SHIFT+SPACEBAR (xem hình dưới)   
  
* Khi để con trỏ chuột trên tên hàm đang soạn thảo, nó cũng sẽ hiện thị thông tin cơ bản về hàm.

