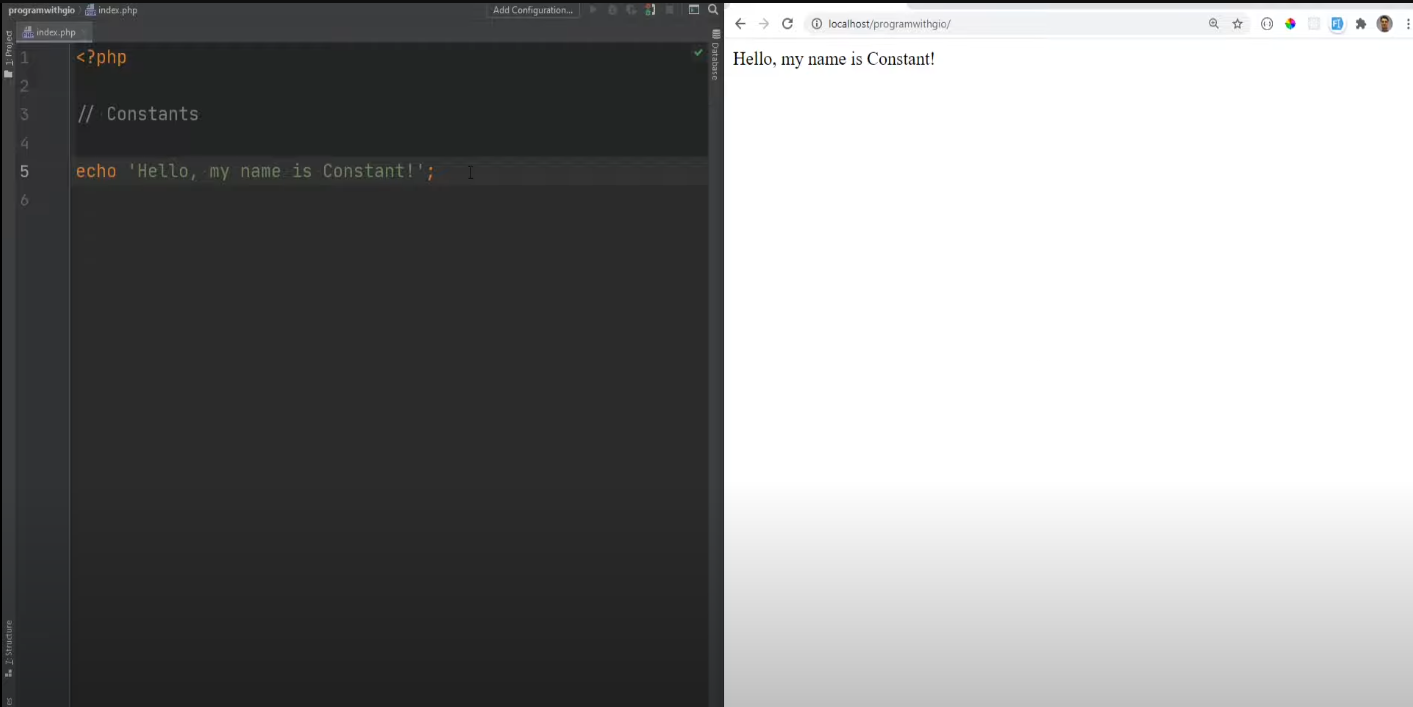
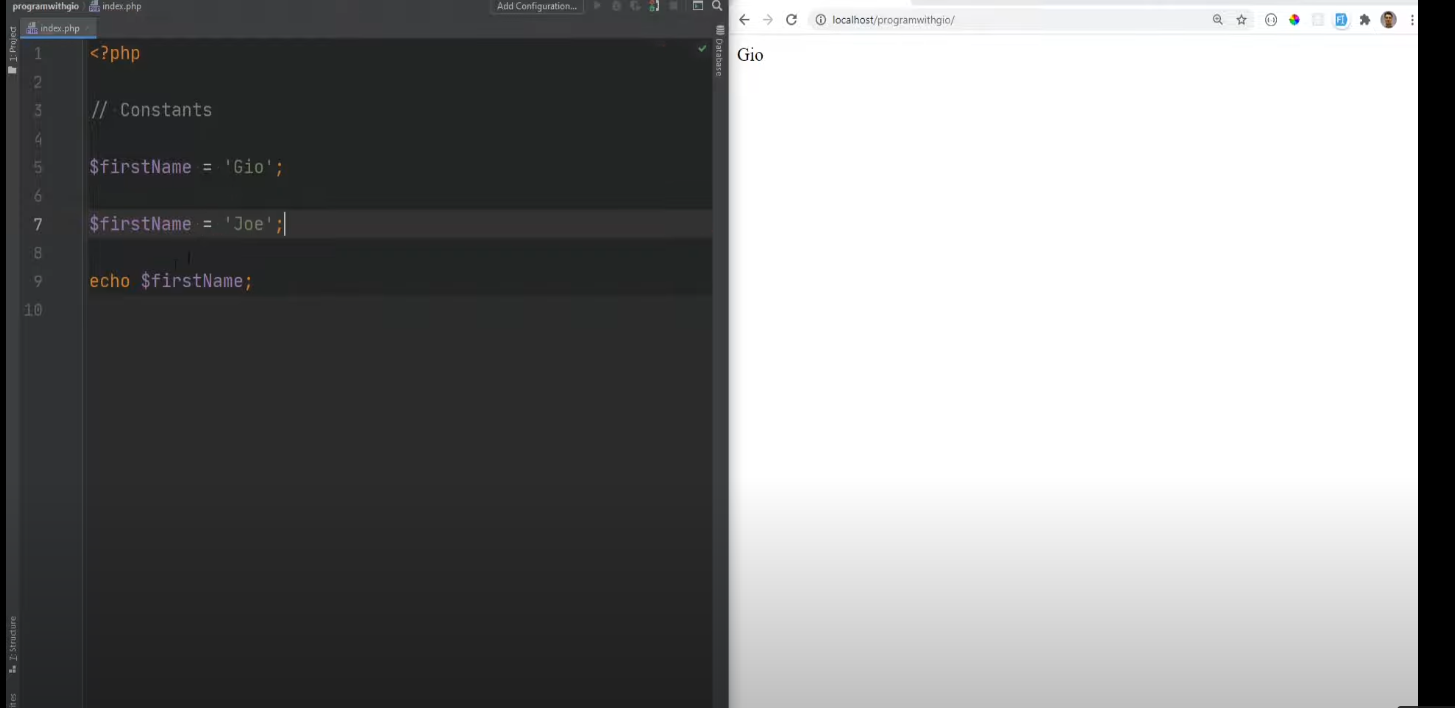
**1.3 What Are Constants & Variable Variables In PHP - Full PHP 8 Tutorial**

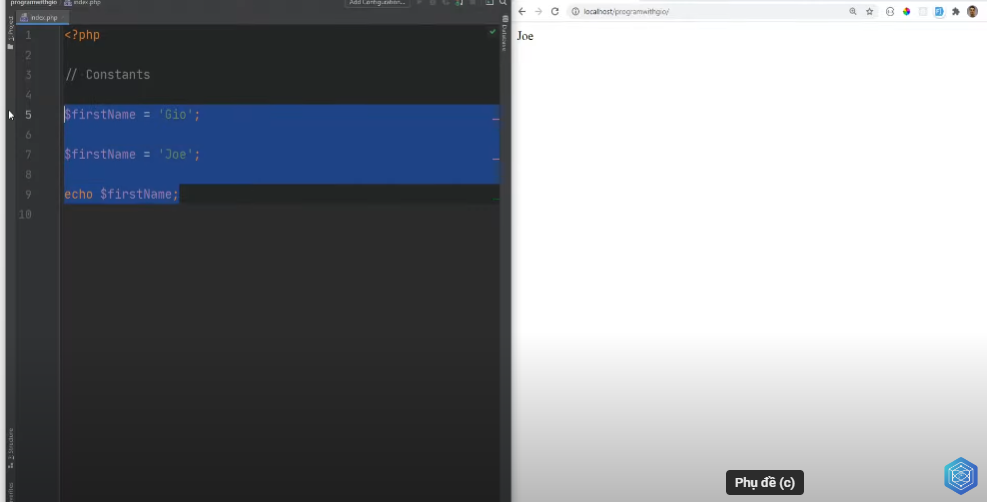
\* Hãy nói về **hằng số - hằng số** là các giá trị không thể thay đổi một khi chúng được xác định, không giống như các biến mà bạn có thể xác định và ghi đè các biến thô sau khi chúng được xác định

* Ví dụ như trong video trước chúng ta đã học cách xác định biến, vì vậy hãy thực hiện nhanh chóng



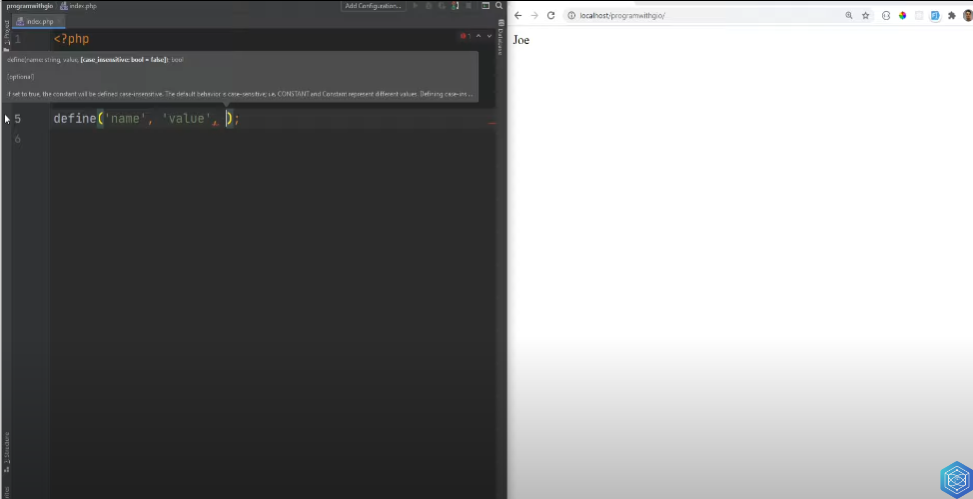
* Ta sẽ đặt tên đầu tiên và sau đó lặp lại tên đó , bây giờ chúng ta có thể ghi đè giá trị này ngay bên dưới và thay đổi nó thành tên khác.
* Trong trường hợp này **Joe** sẽ được in trên màn hình chứ không phải **Geo**
* Vì **PhP** là được thực thi từ dòng trên xuống dưới theo từng dòng nên đầu tiên nó đọc giá trị này và gán giá trị **Geo** cho tên, sau đó nó gán lại tên cho **Joe** và sau đó nó in **Joe** thay vì **Geo**





\* Đối với các **hằng số** khi bạn xác định chúng, bạn không thể ghi đè hoặc thay đổi giá trị.

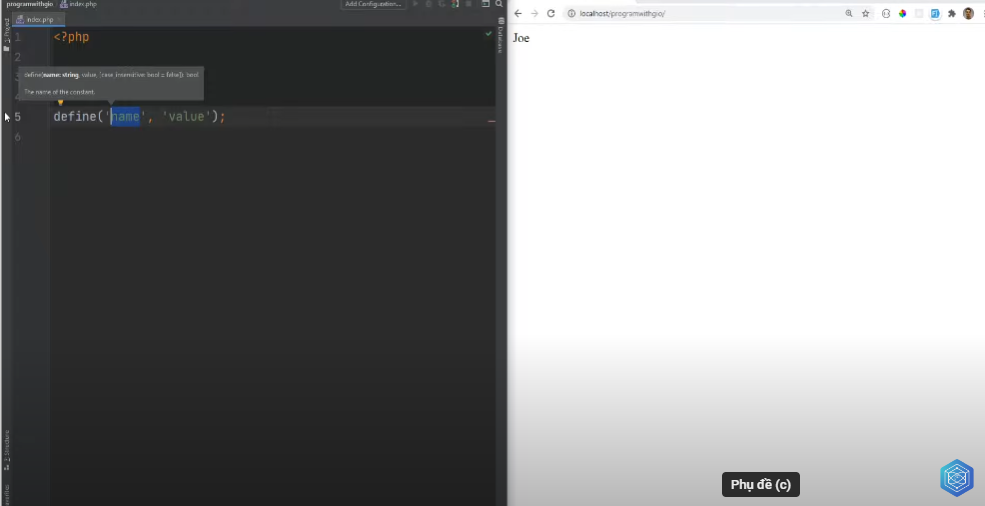
* Có **2 cách** có thể xác định **hằng số** và chúng ta sẽ thực hiện cả 2 cách để xác định hằng số bằng cách sử dụng **hàm** có tên là **định nghĩa**
* Đừng lo lắng về các hàm, ngay bây giờ chúng ta sẽ xem xét các chức năng chi tiết hơn trong các video sau nhưng đây là cách bạn **định nghĩa một hằng số** mà gọi là xác định và xác định chấp nhận hai tham số.



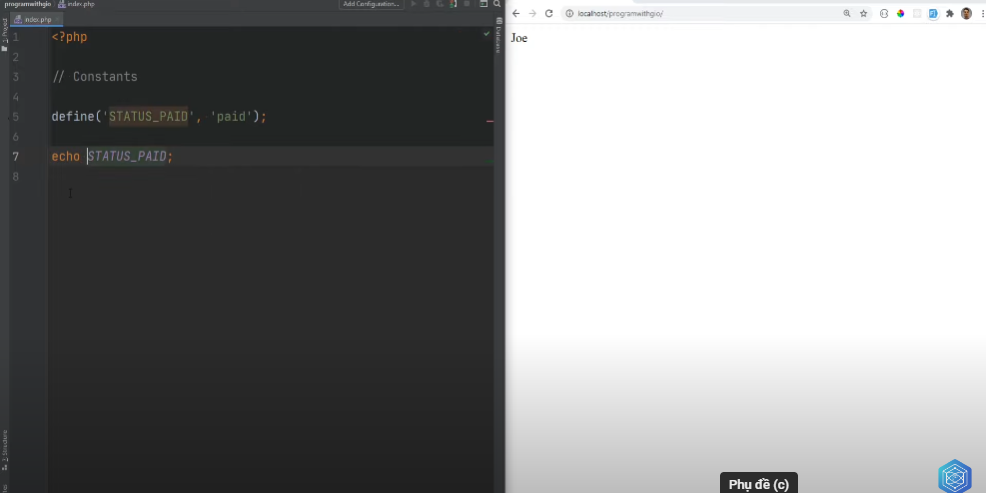
\* **Tham số đầu tiên** là tên và **tham số thứ hai** là giá trị

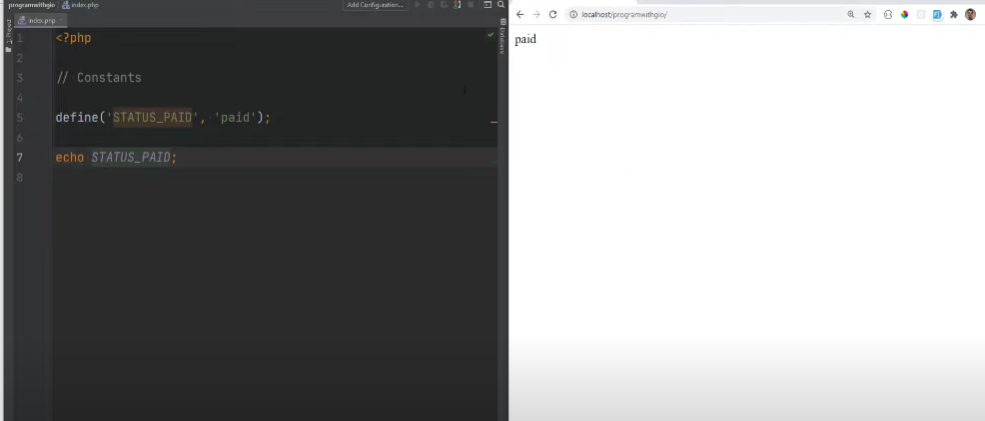
\* Bây giờ bỏ qua **tham số thứ ba** ở đây, về cơ bản cho phép bạn đặt đúng hoặc sai cho dù tên hằng của bạn có phân biệt chữ hoa chữ thường hay không, điều này không được dùng nữa và không nên sử dụng nó như vậy.

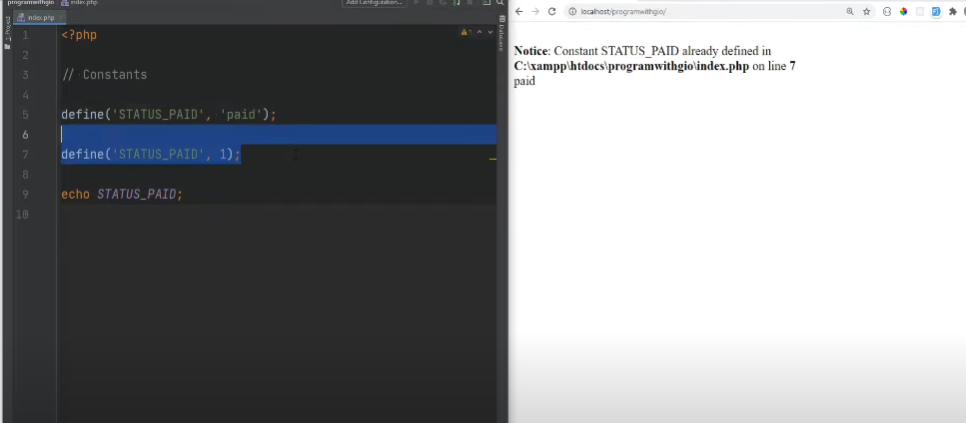
* **Tên** của **hằng** tuân theo các quy tắc tương tự như **biến**
* Nghĩa là nó chỉ có thể bắt đầu bằng các chữ cái hoặc dấu gạch dưới và bạn **không thể chứa** bất kỳ ký tự đặc biệt nào
* Tên của các **hằng** cũng phân biệt chữ hoa chữ thường giống như các biến
* Vì vậy bạn nên sử dụng chúng theo cách bạn xác định chúng. **Quy ước** là bạn xác định **các hằng số** chỉ viết hoa và bạn chỉ cần tuân theo quy ước đó và tuân theo nó.

****

* Chúng ta sẽ xác định một **hằng số** được gọi là trạng thái đã thanh toán và đặt giá trị thành đã thanh toán, sau đó bạn chỉ cần tham chiếu hằng số đó bằng cách sử dụng tên của hằng số để chúng ta có thể lặp lại trạng thái đã thanh toán và điều đó sẽ có tác dụng

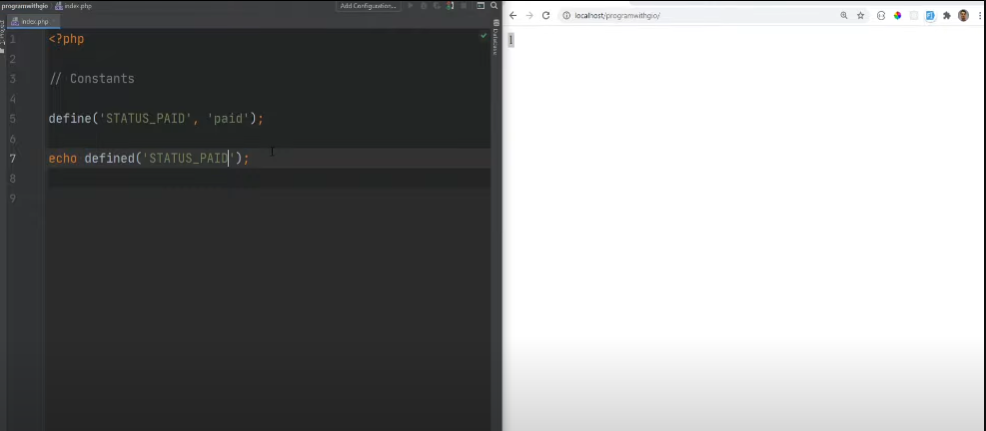
****

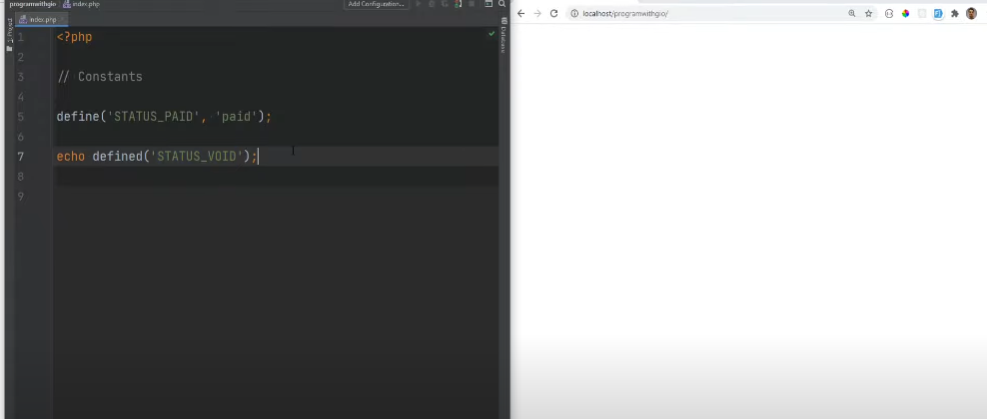
* Ta sẽ **không thêm** ký hiệu đô la ở đây giống như bạn làm với các biến
* Vì khi tham chiếu các hằng số bạn không cần ký hiệu đô la ****
* Như đã đề cập trước đó, bạn **không thể** thay đổi giá trị của hằng số nên bạn không thể xác định ngay bên dưới, điều này sẽ gây ra lỗi cho bạn

****

**\***Bạn cũng có thể **kiểm tra** xem một **hằng số** đã được xác định hay chưa bằng cách sử dụng **hàm** đã xác định.  
\*Đừng lo lắng về **hàm** đó như tôi đã nói trước khi chúng ta đề cập chi tiết hơn về sau nhưng đây là cách bạn sẽ kiểm tra xem **hằng số** đã được xác định hay chưa

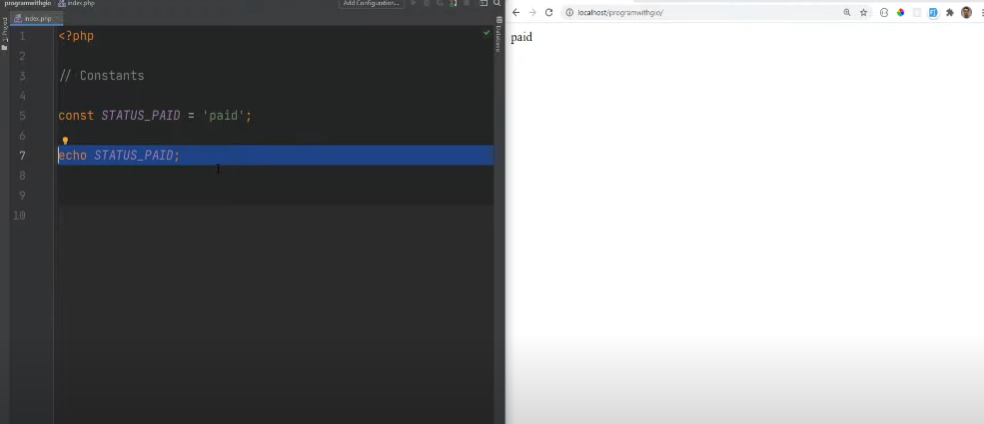
* Bạn có thể thực hiện xác định **echo** và sau đó đặt tên của **hằng số** được thanh toán trạng thái và sau đó chúng tôi sẽ in một **hằng số** này về cơ bản trả về **boolean** true hoặc false
* Chúng tôi sẽ chạm vào **booleans** và các **kiểu dữ liệu** trong video tiếp theo nhưng giá trị **boolean** true được in dưới dạng một trên màn hình
* Đó là lý do tại sao chúng ta thấy **1** ở đây

****

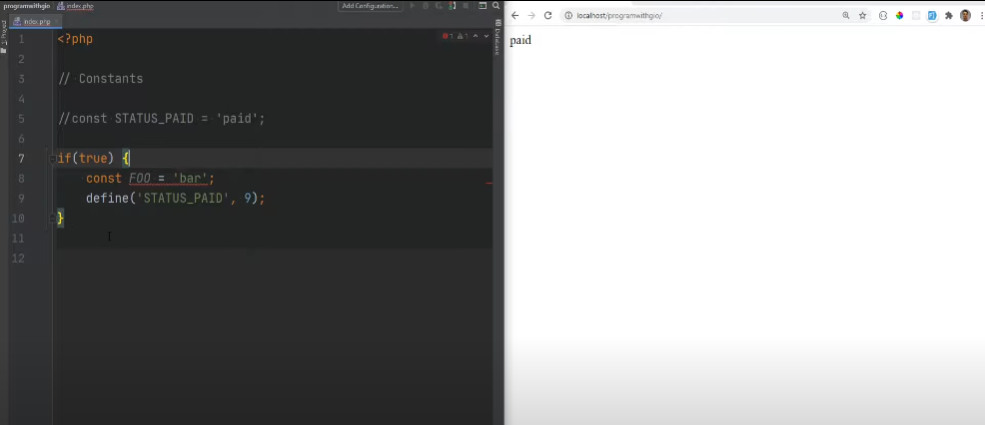
* Nếu chúng ta **kiểm tra** trạng thái khác, chúng tôi thấy rằng không có gì được in, điều đó có nghĩa là **zetas void** không được xác định 

\*Một cách khác để xác định **hằng số** là sử dụng từ khóa **const** để chúng ta có thể thực hiện điều tương tự bằng cách sử dụng từ khóa **const**

* Để chúng ta có thể thực hiện **trạng thái** đã thanh toán và sau đó lặp lại theo cách tương tự và cách này hoạt động.
* Điểm khác biệt chính giữa hàm định nghĩa và từ khóa **const** có phải là các hằng số được tạo bằng từ khóa **const** thực sự được xác định tại thời điểm biên dịch trong khi các **hằng số** được tạo bằng hàm định nghĩa được xác định trong thời gian chạy.
* Bạn không thể xác định các **hằng số** bằng từ khóa **const** trong các cấu trúc điều khiển của mình như vòng lặp hoặc nếu không
* Chẳng hạn như bạn có thể định nghĩa các **hằng số** bằng cách sử dụng **hàm định nghĩa** trong cấu trúc điều khiển trong các video sau

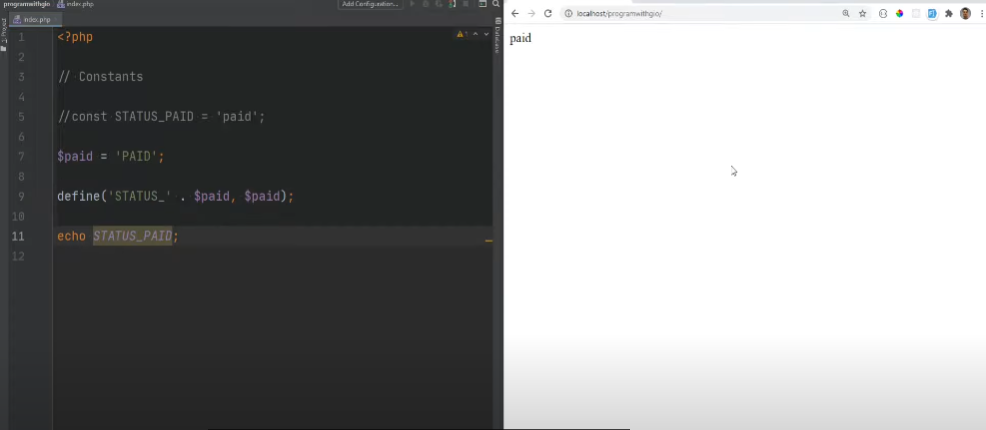
****

* Chúng ta sẽ nói về cấu trúc điều khiển trong các video sau
* Nhưng nó có thể làm điều gì đó kiểu như nếu đúng xác định **trạng thái** đã thanh toán và chúng ta có thể nhận xét điều này ở đây và điều này sẽ hoạt động tuy nhiên sẽ không hoạt động với từ **const**
* Vì vậy nếu tôi thực hiện **const** đầy đủ thanh thì điều này sẽ không hoạt động và **id** sẽ gạch chân nó cho bạn
* Một vài điểm khác biệt nữa nhưng tôi sẽ không nói về tất cả chúng ngay bây giờ vì có một số chủ đề nâng cao trong đó và chúng ta sẽ đề cập đến những chủ đề đó trong video sau.

****

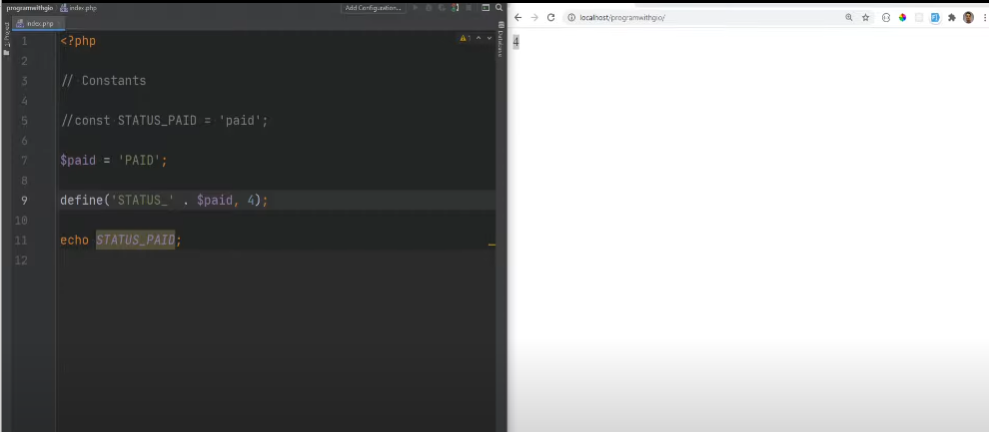
\*Nhưng có một điểm khác biệt nữa mà tôi muốn cho bạn xem:

* Chẳng hạn như bạn có thể sử dụng các **biến** để xác định tên **hằng số** của mình
* Chẳng hạn như xác định dấu gạch dưới **trạng thá**i và sau đó đặt **giá trị** này thành **giá trị** đã thanh toán rồi đặt **giá tr**ị thành đã thanh toán
* Sau đó bạn có thể đặt **biến** đó thành trang như thế này và bây giờ bạn có **trạng thái** gạch dưới được xác định mặc dù **id** sẽ đánh dấu nó vì nó chưa được xác định, chúng ta đã tạo nó một cách linh hoạt
* Vì vậy nếu chúng tôi vào đây và in thì nó sẽ in trang vì đó là giá trị của **biến**

****

\* Bạn có thể đặt cái này thành bất cứ thứ gì bạn muốn bạn có thể gửi cái này tới một số nguyên

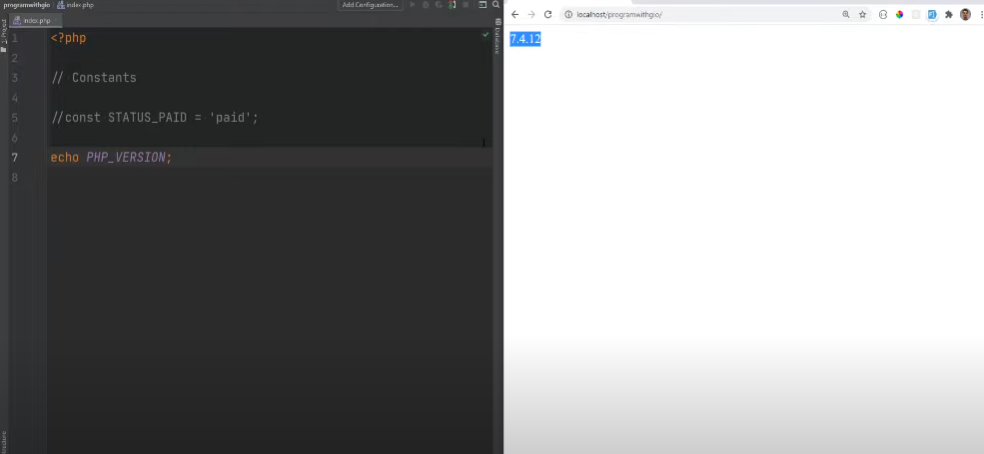
* Nếu bạn muốn và nó sẽ **in 4**
* Vì vậy khi nào bạn sẽ sử dụng **hằng số** bất cứ khi nào bạn có một số **dữ liệu tĩnh** không thay đổi quá thường xuyên,
* Tôi sẽ nói là hãy dùng **hằng số**
* Ví dụ: bạn có thể sử dụng chúng cho các trạng thái như **trạng thái** đã thanh toán, **trạng thái void** đang chờ bị từ chối, v.v.,
* Bây giờ, ngoài các **hằng số** tùy chỉnh mà bạn có thể xác định, còn có một số **hằng số** ma thuật và được xác định trước mà php cung cấp
* Ví dụ: bạn có thể in phiên bản php bằng hằng số được xác định trước

****(((\*Đó là phiên bản gạch dưới của php và bạn có thể thấy có rất nhiều hằng số khác ở đây

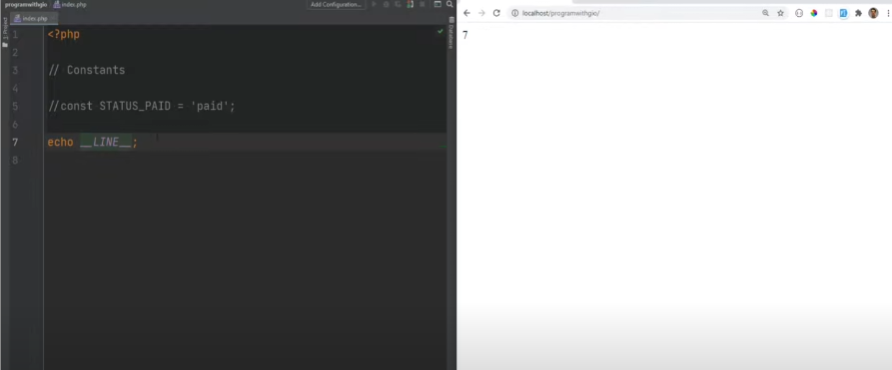
\*Tôi sẽ đăng liên kết trong phần mô tả sẽ bao gồm tất cả các hằng số được xác định trước nhưng liên kết này sẽ in phiên bản php và hiện tại tôi đang sử dụng 7.4 nhưng Tôi sẽ nâng cấp lên php 8 vì nó mới được phát hành gần đây và chúng tôi sẽ sử dụng php 8 cho phần còn lại của khóa học. ))) **# lời nói trong video**

\* Tôi cũng đã đề cập đến các **hằng số** ma thuật và các **hằng số** ma thuật gần như là các hằng số nhưng chúng được gọi là ma thuật

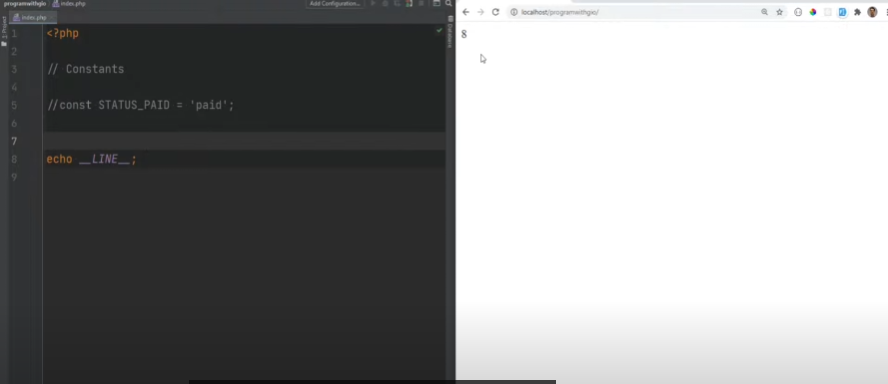
* Vì giá trị của chúng thực sự có thể thay đổi tùy thuộc vào nơi chúng được sử dụng

****

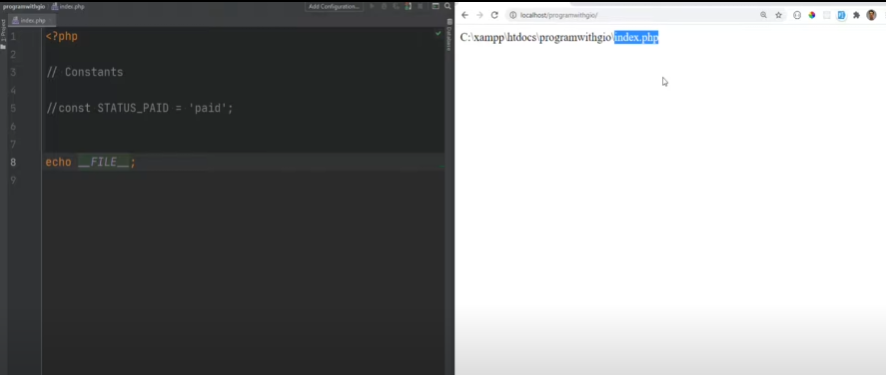
* Ví dụ: Một trong số chúng là dòng gạch dưới và bạn có thể thấy một loạt chúng ở đây, tôi cũng sẽ đưa liên kết đến các hằng số ma thuật này vào phần mô tả để bạn có thể đọc thêm về chúng và phần này chỉ chứa giá trị của số dòng nơi nó được sử dụng nên trong trường hợp này nó được sử dụng ở dòng số bảy nên họ sẽ **in 7**

****

\*Chuyển nó xuống dòng số tám thì bây giờ nó sẽ **in ra số 8**

****

* Tương tự với **Tệp**, nó sẽ in đường dẫn tệp nơi nó được sử dụng
* Vì vậy trong trường hợp này, nó được sử dụng trong **index.php** và nó in đường dẫn đầy đủ và cuối cùng hãy chuyển sang các biến



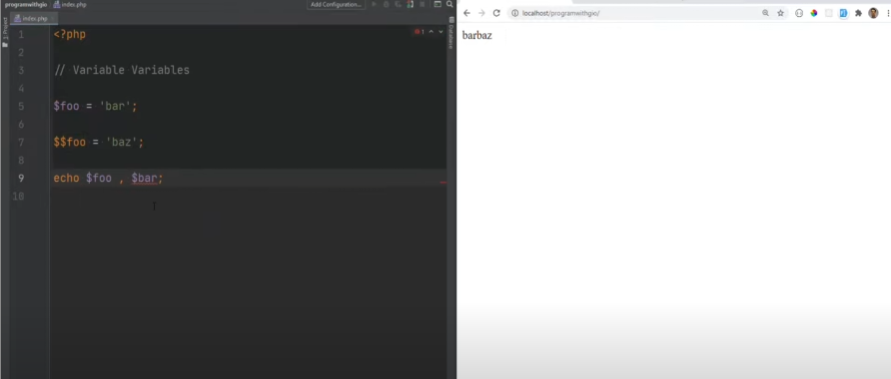
* Vì vậy hãy xác định **một biến** ở đây

\*Để tôi đổi **biến** này thành **biến biến** hãy tạo một **biến** tên là đầy đủ và đặt nó thành **bar** rồi - sau đó chúng ta sẽ đặt **biến biến** bằng cách sử dụng **ký hiệu** đô la và **biến** đó và chúng ta sẽ đặt **biến** đó thành **bus**

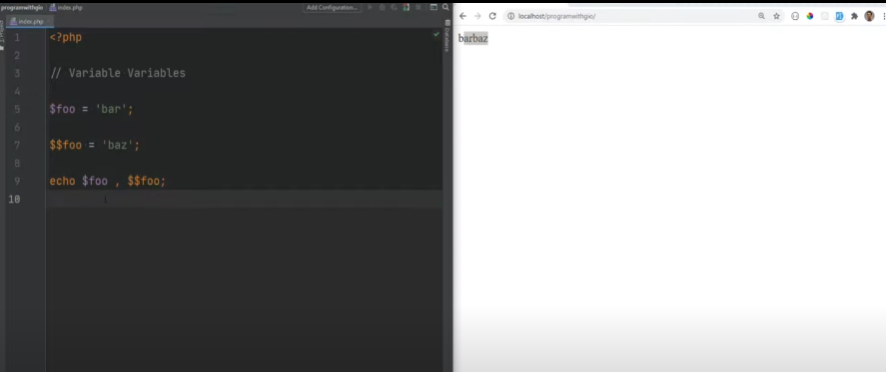
* Tôi sẽ giải thích **biến biến** là gì để về cơ bản, một **biến biến** lấy **giá trị** của **biến** và coi đó là tên của **biến mới**
* Vì vậy trong trường hợp này về cơ bản giống như **thanh bằng** **buzz** vì **bar** là **giá trị** của **biến foo**

\*Về cơ bản chúng tôi đang nói ở đây rằng lấy **giá trị** của **biến foo** và sử dụng giá trị đó làm tên của **biến** mới để nó tạo ra **một biến mới**

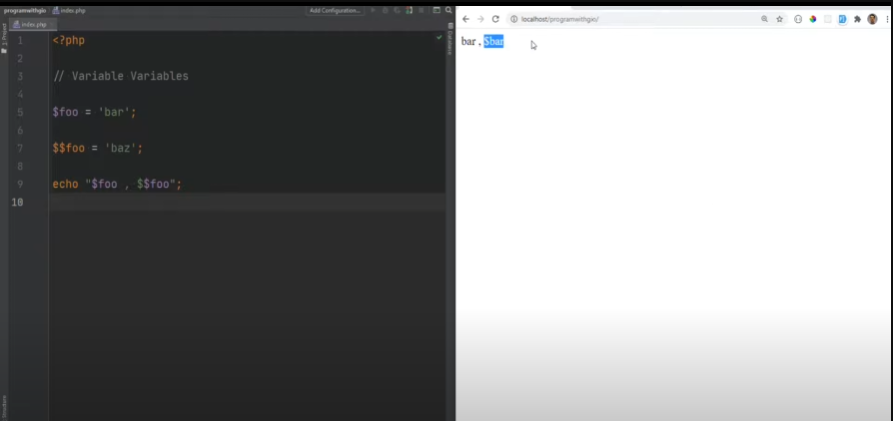
* Cho bạn được gọi là **bar** và sau đó bạn có thể **in** nó ra như thế này **foo** rồi **bar** và cái này sẽ hoạt động ngay cả khi **id** gạch chân cái này và cho bạn biết rằng thanh **chưa được xác định**, nó thực sự **được xác địn**h bằng cách **sử dụng** các **biến biến**

****

* Chỉ là **id** không biết rằng cách thích hợp **khác** để **in** cái này ra sẽ là sử dụng **biến biến** ở đây như thế này và nó sẽ hoạt động theo cách tương tự

****

* Nếu bạn muốn **đặt** cái này **trong dấu ngoặc kép** như thế này thì nó sẽ không hoạt động và nó sẽ chỉ **in thanh biến** như vậy để thực hiện công việc này.

****

* Bạn cần **đặt toàn bộ** nội dung trong ngoặc như thế này và cách này sẽ hiệu quả hoặc **cách khác** là bạn **xóa một ký hiệu đô la** và đặt nó **bên ngoài** dấu ngoặc và điều này cũng có tác dụng và điều này rất hữu ích
* Vì điều này mang lại cho bạn tính linh hoạt để tạo các biến của bạn một cách linh hoạt

