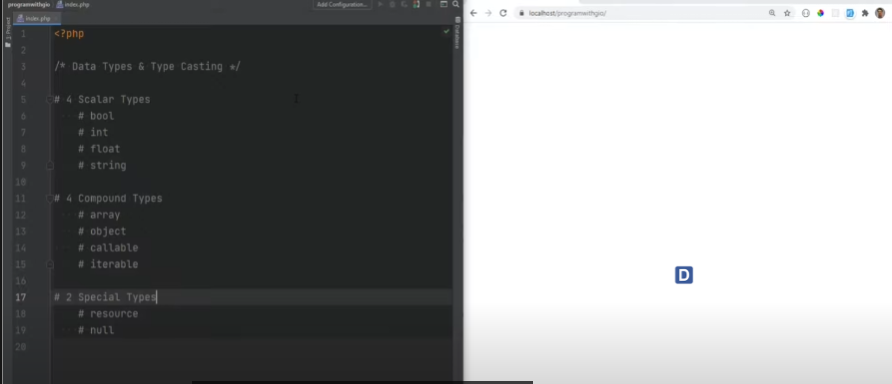
**1.4. PHP Data Types - Typecasting Overview & How It Works - Full PHP 8 Tutorial**

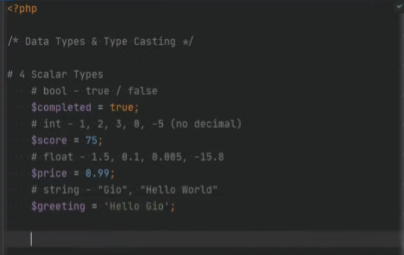
**\***Hãy nói về các kiểu và cách định kiểu trong **PhP**

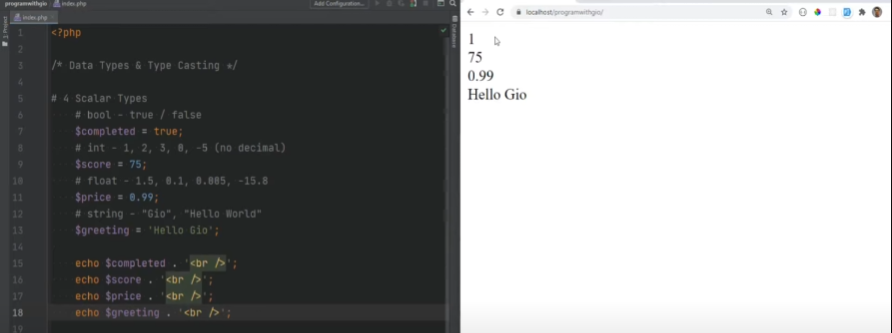
- **PhP** được gõ động hay còn được gọi là **ngôn ngữ** được gõ hàng tuần, trong đó bạn **không bắt buộc** phải xác định loại **biến** của mình và loại **biến** cũng có thể thay đổi sau khi được xác

* Định nghĩa là **ngôn ngữ** được gõ động rằng việc kiểm tra kiểu xảy ra trong thời gian chạy trong khi **ngôn ngữ** được **gõ tĩnh** có nghĩa là việc kiểm tra kiểu xảy ra vào **thời gian** biên dịch . Ví dụ java c++ và c Sharp là các ngôn ngữ được **gõ tĩnh**
* Vì **PhP** cho phép hệ thống kiểu như vậy linh hoạt hơn nhưng tính **linh hoạt** đó có giá **hiệu năng** và đôi khi có thể gây ra các lỗi không mong muốn tuy nhiên **PhP** đã cải thiện hệ thống kiểu của nó rất nhiều trong các phiên bản mới nhất và thậm chí nó còn hỗ trợ các kiểu nghiêm ngặt **PhP** hỗ trợ 10 kiểu nguyên thủy được nhóm theo **4 kiểu vô hướng**, **4 kiểu ghép và 2 kiểu đặc biệ**t và có cũng có **2 loại giả chủ yếu** được sử dụng để dễ đọc, chúng được trộn lẫn và không có giá trị nhưng đừng lo lắng về hai loại này.

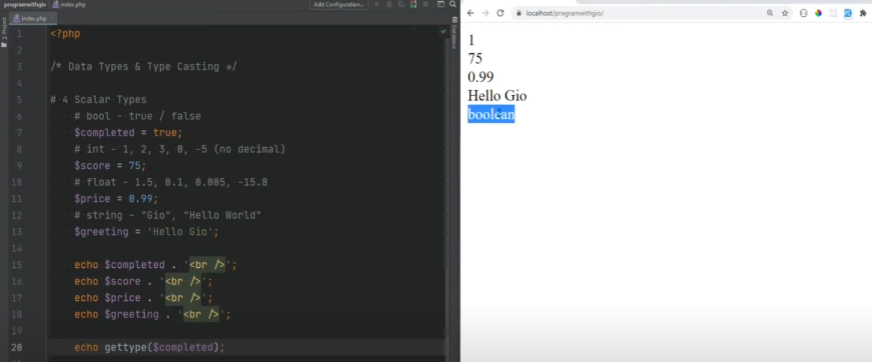
****

\* Vì vậy hãy xem lại những loại này là gì?

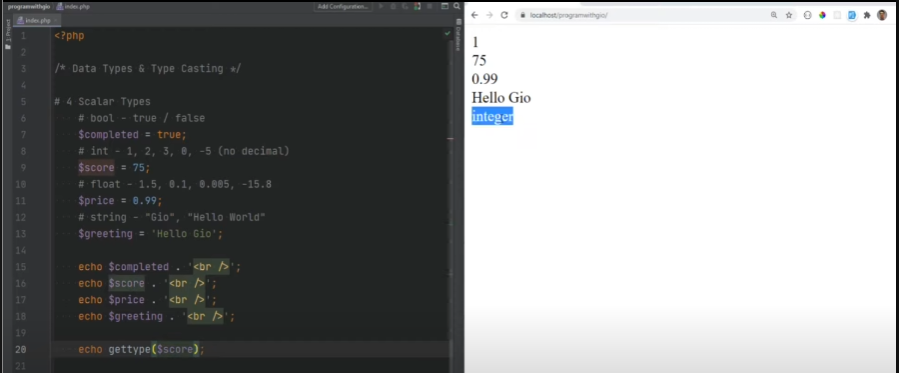
* **Bốn loại vô hướng** là số nguyên boolean float và chuỗi boolean chỉ là một biểu diễn đơn giản của giá trị đúng
* Nó có thể là số nguyên **đúng hoặc sai** là các **giá trị số** không có dấu thập phân, còn được gọi là **số nguyên**
* Vì vậy các ví dụ có thể giống như 1 2 3 0 âm 5, v.v. trên số float hoặc số dấu phẩy động còn được gọi là số nhân đôi hoặc số thập phân là các giá trị số có dấu thập phân
* Ví dụ về số float sẽ là 1,5 0,1 0,005 âm 15,8, v.v.
* Đây chỉ là một chuỗi ký tự được đặt trong dấu ngoặc đơn hoặc dấu ngoặc kép nên đây có thể là Geo hello world, v.v.,
* Hãy xác định một số loại này là biến và in chúng ra màn hình - Hãy tiếp tục và xác định một biến có tên đã completed và đặt giá trị thành true hãy xác định một biến có tên là Point và đặt nó thành 75 - Hãy xác định một biến có tên là Price và đặt nó thành 0.99 xu - Hãy xác định một biến có tên là greeting và nói Hello World  ****
* Vì vậy để in từng cái trong số chúng ra một dòng mới   
  - Ta sẽ **thêm br của html** vào dòng lệnh để **in ra** các dòng riêng biệt . Bấm làm mới trang để hiện ra 3 giá trị dưới hiện lên có ý nghĩa còn **giá trị** đầu tiên không có ý nghĩa .
* Khi in ra true nó sẽ được in thành một và khi in ra sai, nó sẽ được in ở dạng trống nên sai sẽ được in dưới dạng không có gì, điều này không thực sự chỉ ra loại biến.

****

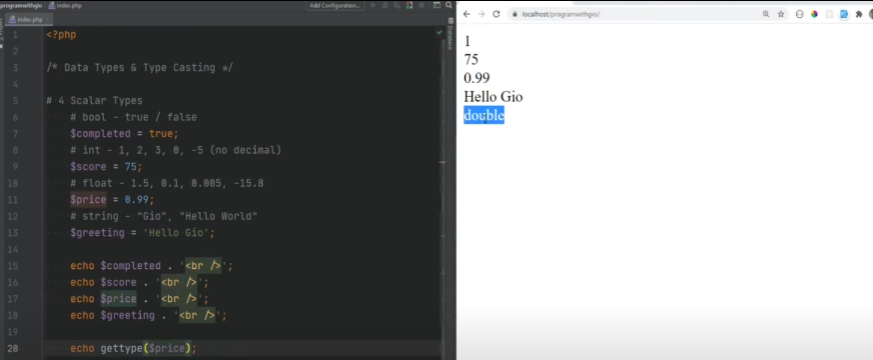
* Có một thứ gọi là **hàm get type** và bạn có thể truyền cho nó **một biến** và nó sẽ trả về kiểu của biến
* Vì vậy, ví dụ ta có thể thực hiện **echo get type** và do done và điều đó sẽ trả về Boolean

****

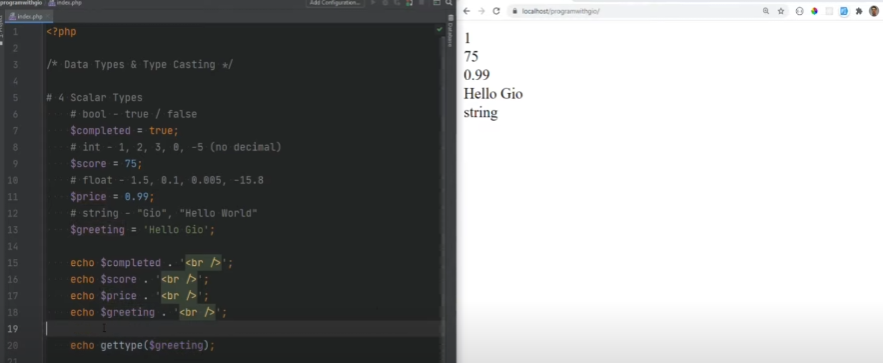
* Chúng ta có thể **score** và họ sẽ trả về **integer**

****

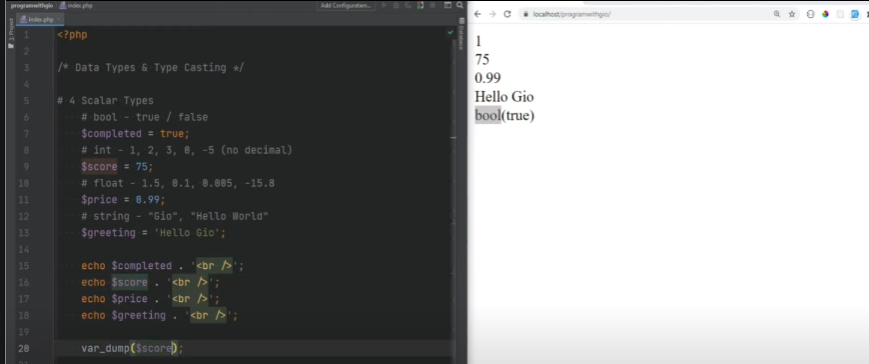
* Chúng ta có thể làm điều đó với **price** và chúng tôi sẽ trả lại **double**, giống như **float**

****

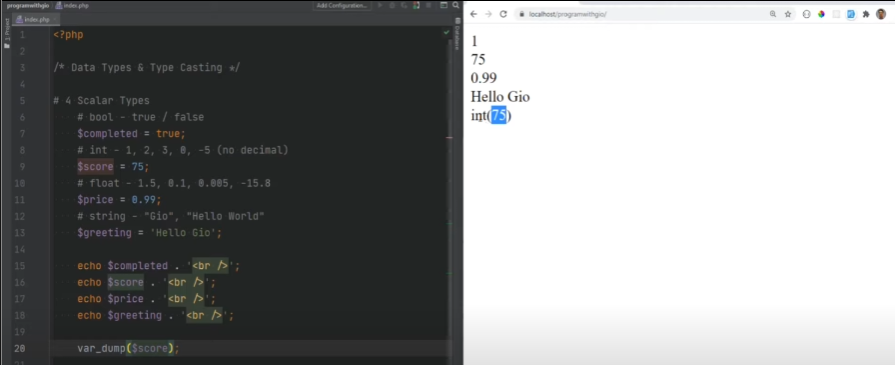
* Chúng ta có thể **greeting** và điều đó sẽ trả về **string**

****

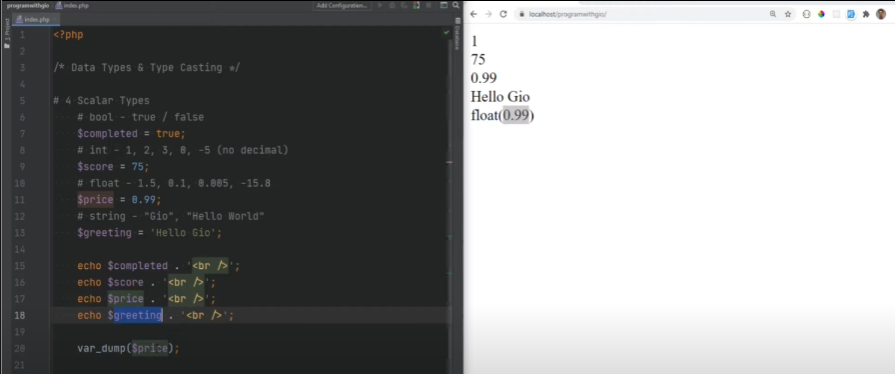
* Một cách khác để tìm ra loại **biến** là sử dụng thứ gọi là **var\_dump** và sau đó bạn cung cấp biến đã hoàn thành trong trường hợp này và về cơ bản.
* **Var\_dump** chỉ in mọi thứ anh ta biết về **biểu thức** bạn **cung cấp** cho nó nên chúng tôi đã đưa ra **biến** hoàn chỉnh ở đây và nếu ta làm mới trang, nó sẽ in **giá trị** mà **biến** đó giữ
* **Giá trị** này đúng trong trường hợp này và nó cũng in **kiểu dữ liệu** của **biến** đó là **boolean** trong trường hợp này



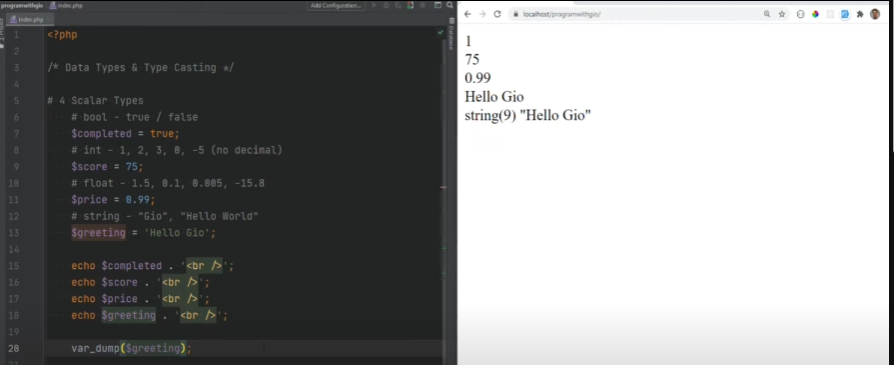
* Nếu chúng ta **thay đổi điểm**, chúng ta sẽ thấy **giá trị** đó là 75 và **kiểu dữ liệu** của nó là số nguyên

****

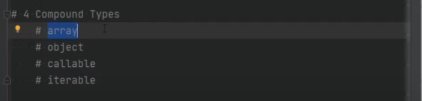
* Hãy giảm giá xuống, **kiểu dữ liệu** là float và giá trị là 0,99

****

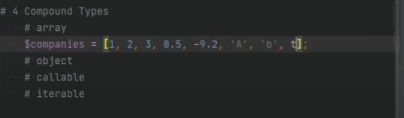
* Hãy **bỏ qua** việc phân loại và đó là **chuỗi kiểu dữ liệu** có 9 ký tự và đây là giá trị

****

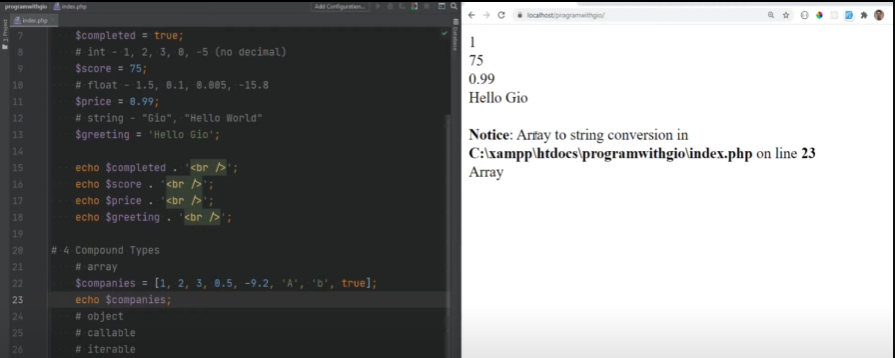
* Vì vậy, đây là những điều cơ bản về **bốn loại vô hướng** như tôi đã đề cập trước khi đi vào **chi tiết hơn** về **từng loại dữ liệu**
* Vì có một số thông tin quan trọng mà bạn cần biết làm thế nào những giá trị này thực sự được chuyển thành các loại khác nhau và cách chúng được xử lý bởi **PhP**
* **Mảng PhP** về cơ bản chỉ là một danh sách các mục và các mục này có thể có nhiều loại khác nhau
* Ví dụ bạn có thể có danh sách các số nguyên bạn có thể có danh sách các boolean bạn có thể có danh sách các boolean hỗn hợp, số nguyên và số float, v.v.

****

* Bạn có thể đặt các **giá trị** vào đây để chúng ta có thể đặt một số thứ như một hai và ba và sau đó chúng ta cũng có thể đặt các số thực 0.5 âm 9.2
* Chúng ta cũng có thể đặt **các chuỗi** vào đó và chúng ta cũng có thể đặt các **booleans** để bạn có thể thấy **mảng** chỉ là **danh sách các mục**

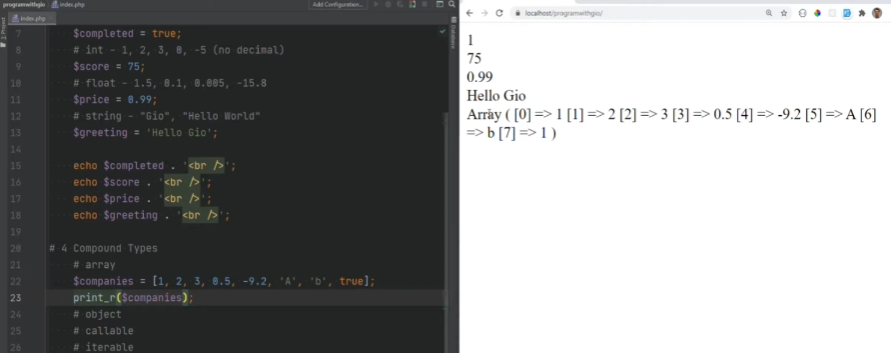
****

* Làm thế nào để bạn **in mảng** nếu bạn thực hiện **echo companies**, nó sẽ chỉ **in mảng** từ
* Nhận thông báo rằng **PhP** đang cố gắng **chuyển đổi** một **mảng** thành **chuỗi**
* Nó gần như **thất bại** vì nó **không biết cách** để **chuyển đổi** một **mảng** thành một **chuỗi** và nó hiện chỉ hiển thị **mảng** từ trên máy tính của bạn
* Điều này có thể bị ẩn tùy thuộc vào **báo cáo lỗi** được đặt trên tệp cấu hình **PhP** của bạn , v.v.

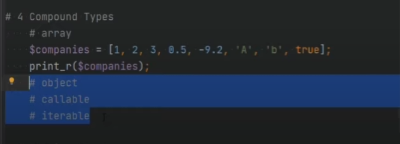
****

* Hiện tại, có một cách khác để bạn có thể in mảng là sử dụng hàm có tên print gạch dưới r
* Bạn có thể thấy, nó chỉ in ở định dạng dễ đọc hơn .

((( Tôi sẽ không đi sâu vào chi tiết hơn về nó trong video này vì tôi sẽ có một video riêng để xóa và tôi sẽ thảo luận về tất cả các chi tiết cần thiết trong đó))) **# lời nói video**

****

((( 3 loại đối tượng ghép cuối cùng có thể gọi và lặp lại là những loại hơi nâng cao nên tôi không muốn làm bạn choáng ngợp với quá nhiều thông tin trong video này nhưng tôi chắc chắn sẽ có video dành riêng cho từng loại này, đừng lo, chúng tôi sẽ đến đó và bạn sẽ biết những loại này là gì và cách sử dụng chúng ))) **# lời nói video**

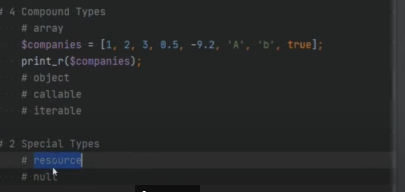
****

* Cuối cùng chúng ta có **hai loại đặc biệt**:

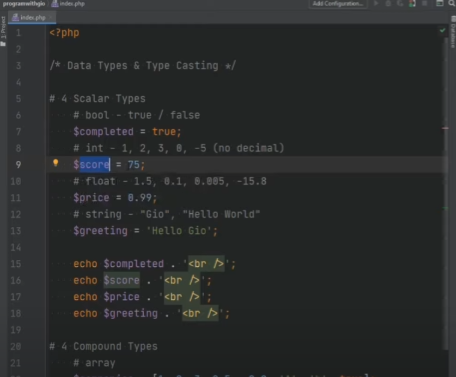
+ một là tài nguyên

+ Loại còn lại là không dành cho tài nguyên, áp dụng tương tự như ba loại này

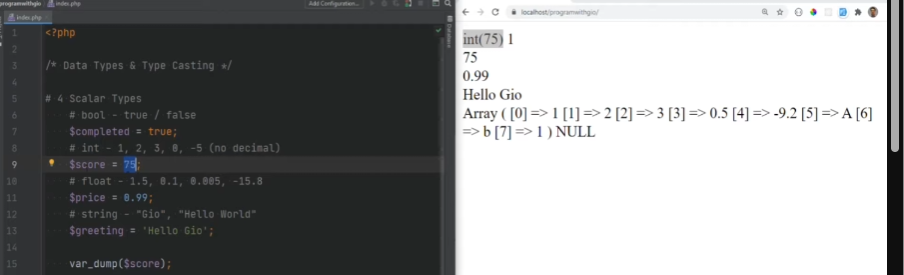
((( Tôi sẽ có một video riêng về nó nên bây giờ đừng lo lắng về nó ))) **# lời nói video**

****

* Không có nó chỉ đơn giản có nghĩa là không có **giá trị** hoặc không có gì nên bây giờ bạn đã biết kiến thức cơ bản về các **kiểu dữ liệu** trong **PhP**
* Làm sao **PhP** thực sự biết rằng biến này chẳng hạn là **một số nguyên**, chúng tôi chưa tuyên bố cụ thể rằng **biến** này là **một số nguyên** mà chúng tôi chỉ đơn giản gán giá trị 75 cho biến score

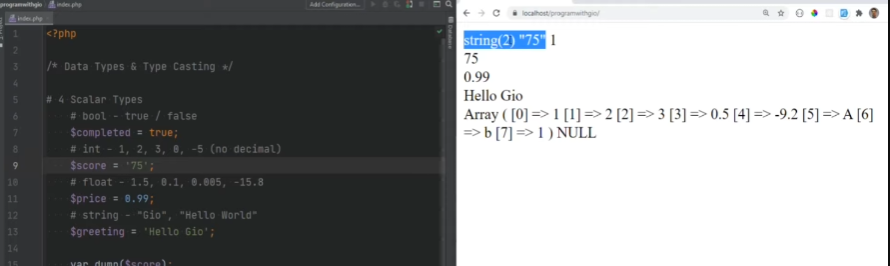
****

* Khi chúng ta thực hiện **var\_dump score**, chúng ta nhận được **số nguyên** và điều này xảy ra
* Vì php sẽ tự động xác định **kiểu dữ liệu** trong thời gian chạy và trong trường hợp này, về cơ bản, **PhP** sẽ nhận ra rằng bạn đang cố gắng xác định **biến** này là **số nguyên** vì nó có không có dấu ngoặc kép xung quanh nó

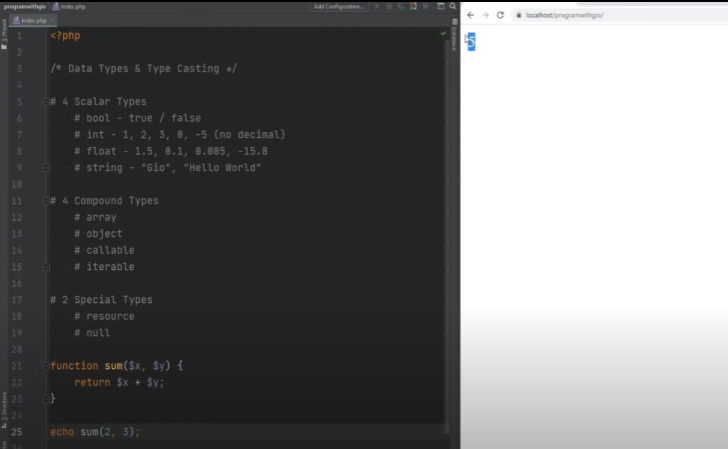
****

* Nếu chúng ta đặt dấu ngoặc kép xung quanh nó thì **PhP** sẽ nhận ra rằng đây không phải là số nguyên
* Đây thực sự là **một chuỗi** và nếu chúng ta thực hiện **var\_dump** như bạn có thể thấy, nó sẽ in 75 với **kiểu dữ liệu** của chuỗi thì cũng có khái niệm về gợi ý kiểu trong các **hàm** và **thuộc tính lớp** của bạn

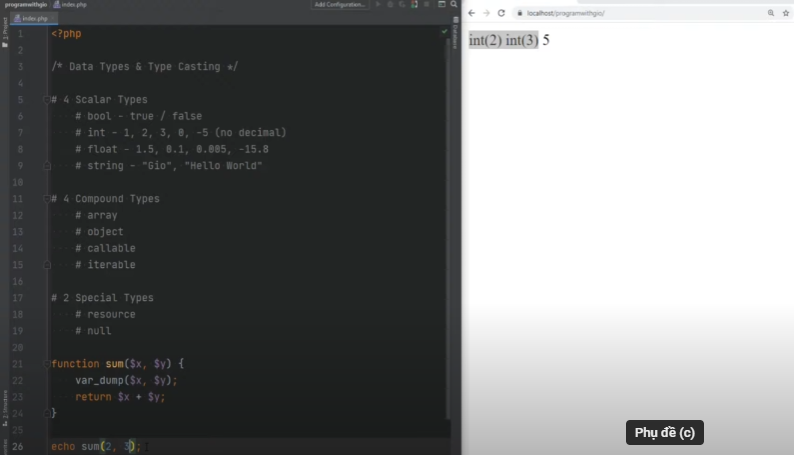
((( Tôi sẽ trình bày chi tiết hơn về các hàm và thuộc tính lớp trong các video riêng biệt, xin lưu ý với tôi ở đây. Tôi chỉ muốn cho bạn xem một ví dụ về gợi ý kiểu là gì và nó hoạt động như thế nào ))) **# lời nói video**

****

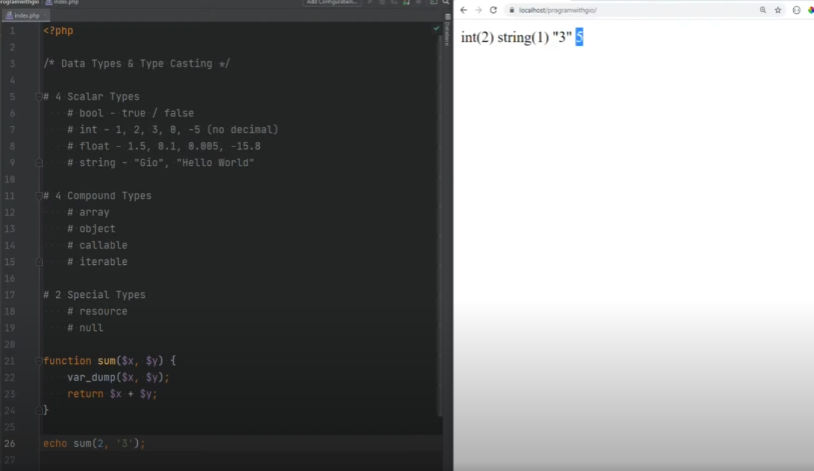
* Ta sẽ tạo **một hàm** tính tổng **hai biến** và trả về **giá trị** nên chúng ta sẽ gọi **hàm sum** chúng sẽ chấp nhận hai tham số x và y và trả về x cộng y
* **Toán tử lớp hoặc kiểu trả** sẽ đề cập đến tất cả những điều này sau ta chỉ đơn giản lặp lại **giá trị** của lệnh gọi **hàm** này nên chúng ta sẽ gọi tổng trên các số 2 và 3. và nếu chúng ta làm mới trang trả về 5.

****

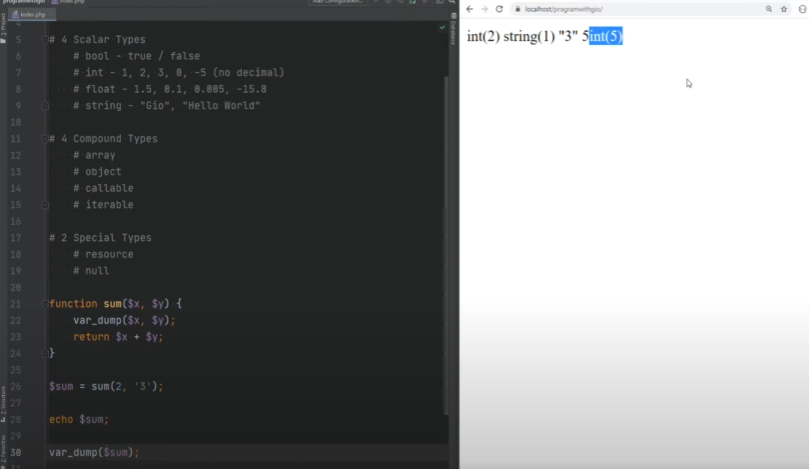
* Vậy là bây giờ nó lại hoạt động như mong đợi **PhP** tự động gán các **kiểu dữ liệu** cho các biến này nếu chúng ta thực hiện **var\_dump** ở đây cho x và y, những **biến** này sẽ xuất hiện dưới dạng **số nguyên**

****

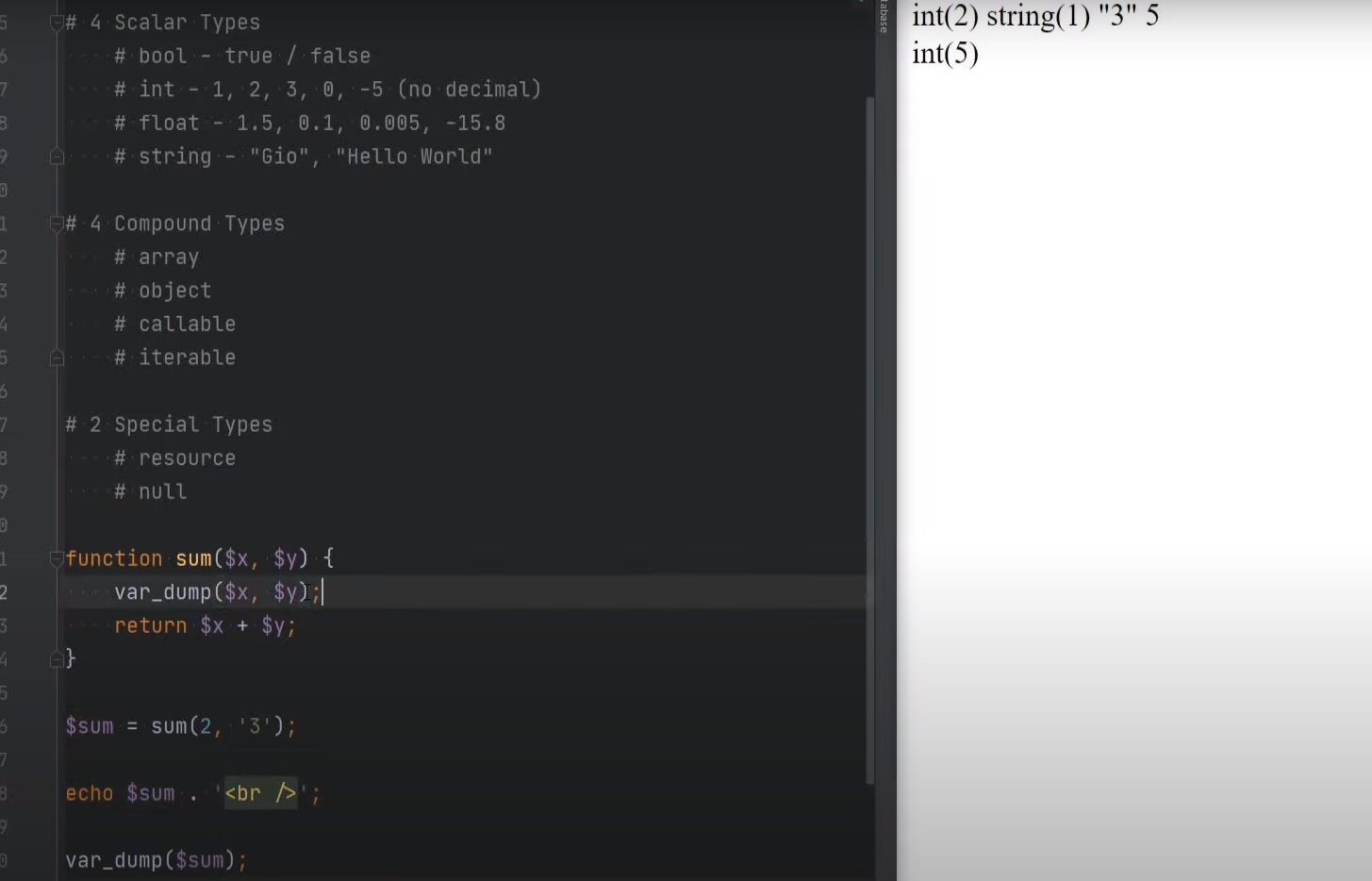
* Nhưng điều gì sẽ xảy ra nếu bạn truyền một cái gì đó giống như **một chuỗi** thay vì **một số nguyên** thì điều gì sẽ xảy ra trong trường hợp đó vì bạn có thể thấy nó lấy **số nguyên** rồi đến **chuỗi** và nó trả về số nguyên 5

****

* Chúng ta biết rằng đó là **một số nguyên** vì chúng ta có thể **gán giá trị** này cho **một biến** rồi tính tổng và sau đó chúng ta cũng có thể var kết xuất tổng ngay tại đây và chúng ta thấy rằng giá trị đó là số nguyên

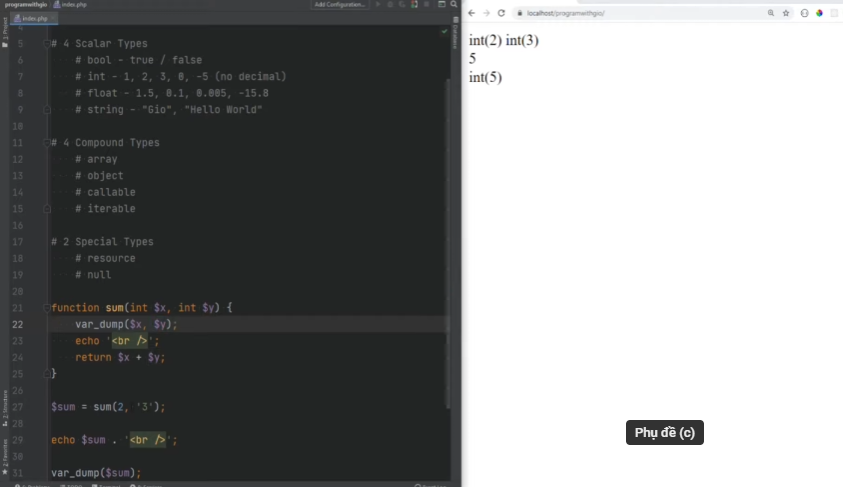
****

* Hãy **thêm br html** ở đây cho rõ ràng và cũng thêm ngắt dòng ở đây

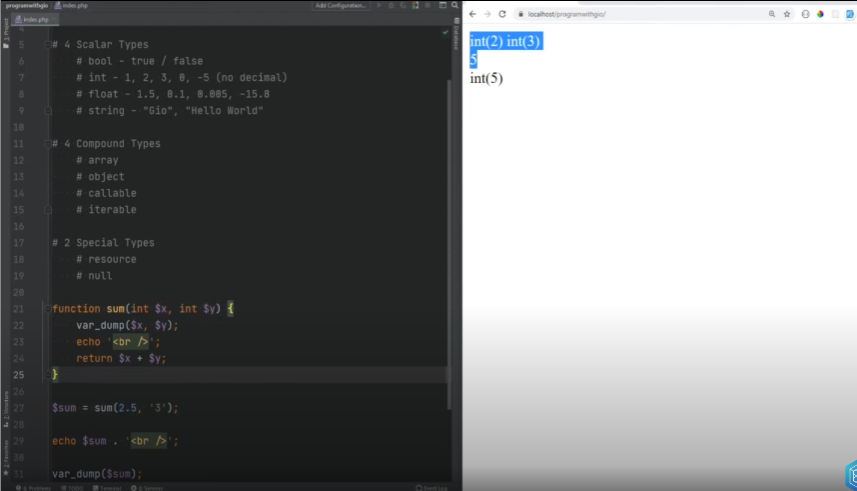
****

\*Bạn có thể chỉ định rằng các **biến** x và y là **số nguyên** và bây giờ khi chúng ta làm mới trang

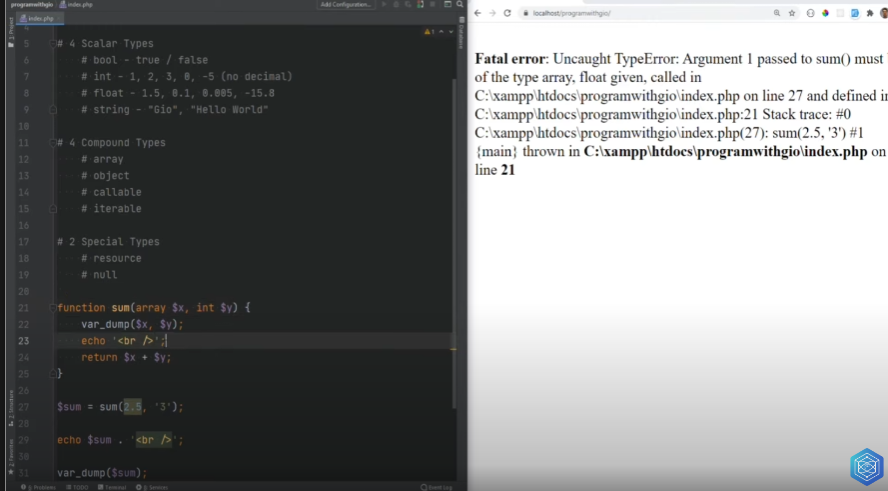
* **Biến thứ hai** sẽ xuất hiện dưới dạng **số nguyên** chứ không phải dưới **dạng chuỗi** mặc dù chúng ta đã **chuyển chuỗi** vào đây
* Đây được gọi là **sắp xếp** kiểu hoặc **ép buộc kiểu** về cơ bản điều đang xảy ra là mặc dù chúng đã gợi ý rằng **biến** này phải là **số nguyên**
* **PhP** vẫn sẽ thử và **chuyển đổi giá trị** của hàm thành **kiểu dữ liệu** thích hợp
* Vì vậy điều này có nghĩa là kiểu của **biến** được xác định bởi ngữ cảnh trong đó biến được sử dụng

****

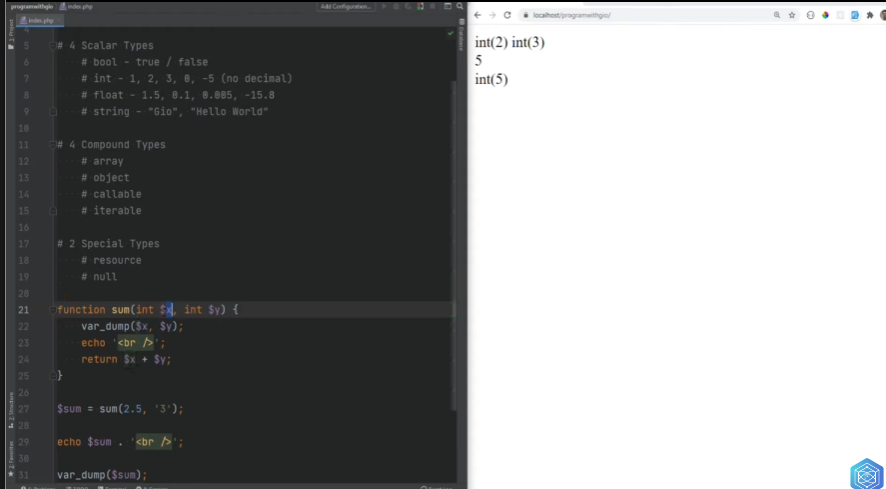
* Bây giờ bạn cũng có thể làm điều gì đó như vượt qua 2.5 dưới dạng float và mặc dù chúng ta đang mong đợi số nguyên nhưng điều này sẽ chuyển đổi số này thành số hai và chúng ta nhận được kết quả chính xác như vậy

****

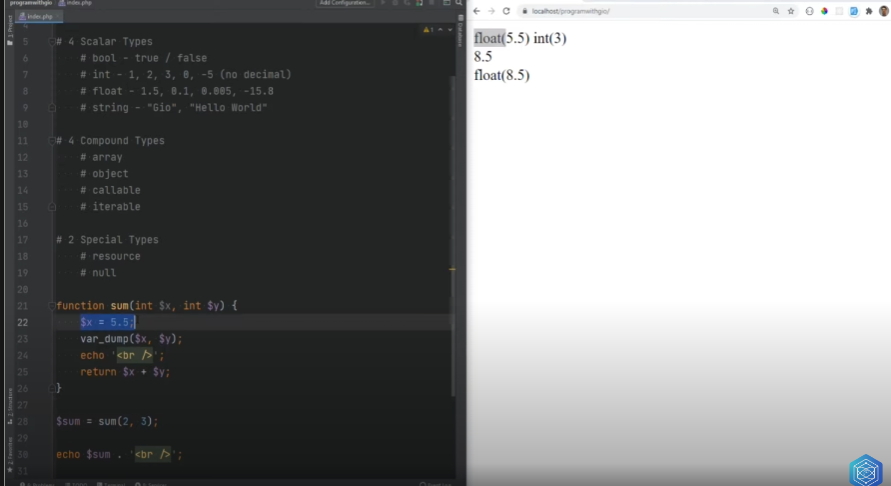
* Nếu trong trường hợp **PhP** không thể thực hiện **chuyển đổi** này một cách **linh hoạt** thì nó sẽ **báo lỗi**
* Chẳng hạn như nếu chúng ta mong đợi một cái gì đó giống như **một mảng** và làm mới trang, chúng ta sẽ **gặp lỗi** nghiêm trọng

****

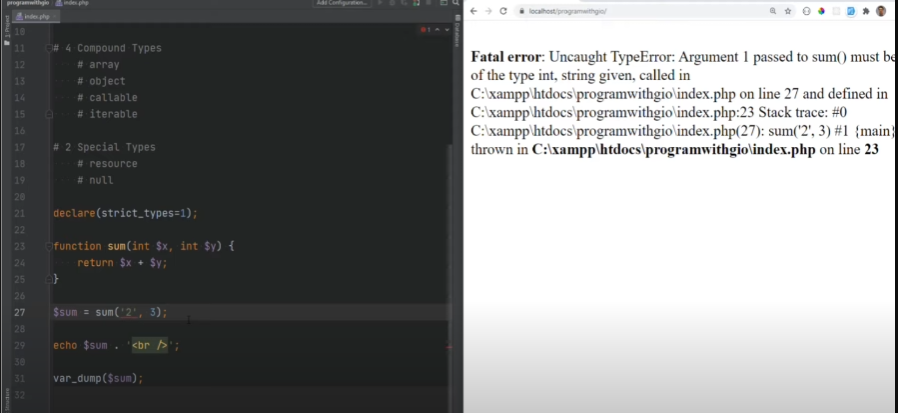
* Điều đáng nói là **loại biến** chỉ được đảm bảo là loại đó cho đến thời điểm đó
* Vì vậy ý tôi là chúng ta được đảm bảo rằng đây sẽ chỉ là **số nguyê**n cho đến thời điểm này

****

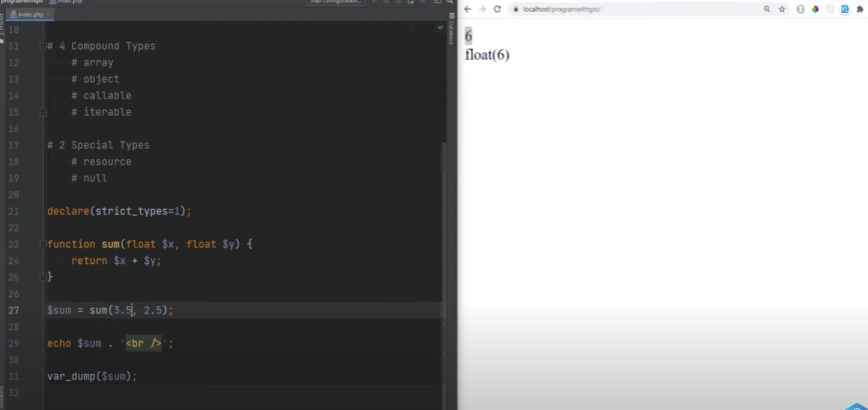
* Có thể đặt ở đây x bằng 5,5 và điều này sẽ chuyển x thành **một số thực** và điều đó hoàn toàn có thể chấp nhận được trong **PhP**
* Vì vậy, ví dụ: nếu chúng ta thay đổi số này thành **số nguyên** và chúng ta thay đổi số này trở lại thành 3 và ta làm mới trang ở đây
* Chúng tôi nhận được 5.5 dưới dạng **float** vì ta đang ghi đè lên 5.5 ở đây nên như đã đề cập trước đó

****

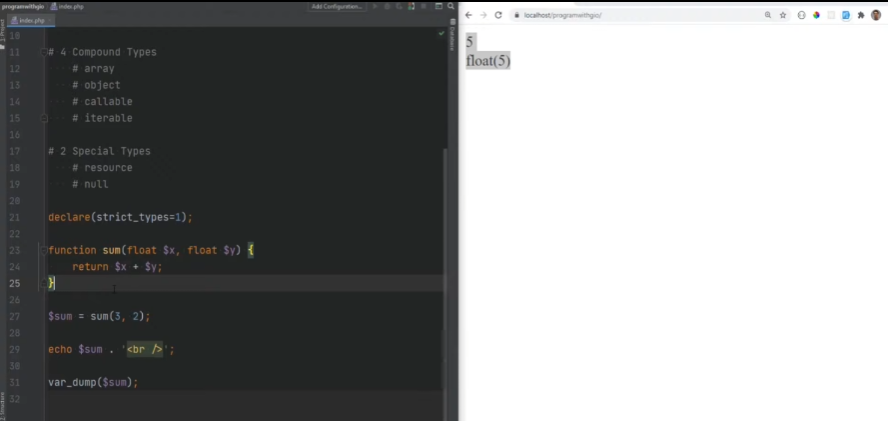
* Vì vậy hãy xóa những thứ này và hãy đảm bảo rằng nó vẫn hoạt động và nó vẫn hoạt động và hãy **thay đổi chuỗi** này **thành chuỗi** và ngay bây giờ nó vẫn hoạt động
* Bây giờ **hãy kích hoạt** loại nghiêm ngặt và bạn có thể kích hoạt nó bằng cách sử dụng **khai báo** các loại nghiêm ngặt bằng một và ngay bây giờ ngay sau khi tôi thêm rằng **id** của tôi đã gạch chân chuỗi này ở đây
* Cho tôi biết rằng **chuỗi** không phải là loại có thể chấp nhận được, nó đang chờ **số nguyên**
* Nếu chúng tôi làm mới trang ở đây thì chúng tôi sẽ gặp phải **lỗi nghiêm trọng** có **một ngoại lệ** ở đây ngay cả trong **chế độ nghiêm ngặt**, bạn vẫn có thể truyền **một số nguyên** trong đó có **kiểu float** mà không có bất kỳ lỗi nào

****

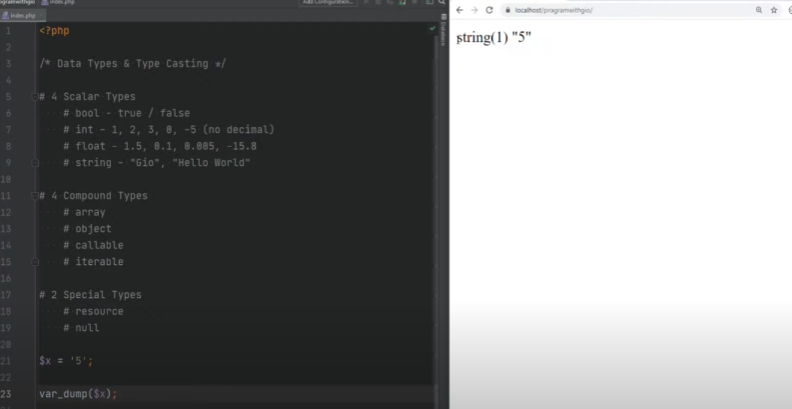
* Vì vậy, ví dụ: Hãy thay đổi các loại dự kiến ​​này từ số nguyên thành số **float** và chuyển **hai số thực** 3,5 và 2,5 và điều này sẽ hoạt động

****

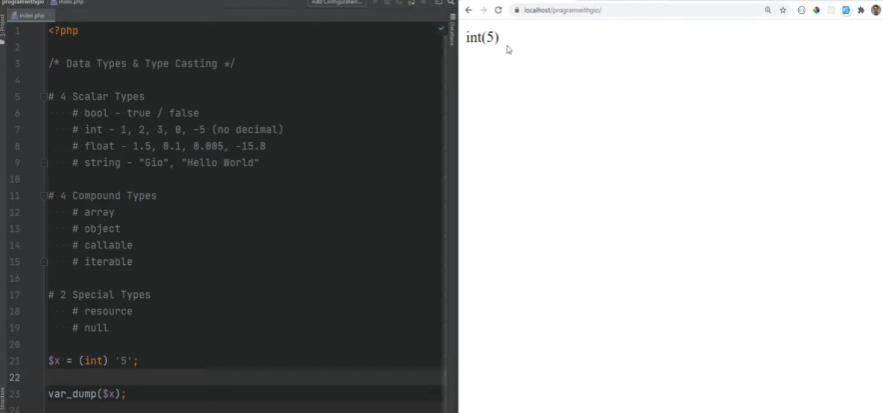
* Hãy chuyển **hai số nguyên** và như bạn có thể thấy **id** không được gạch chân và nếu chúng tôi làm mới thì tính năng này cũng hoạt động tốt
* Vì vậy có nên sử dụng các loại nghiêm ngặt và gợi ý loại hay không?
* Điều đó **hoàn toàn tùy thuộc vào bạn** mặc dù cá nhân tôi khuyên bạn nên sử dụng và cá nhân tôi sử dụng gợi ý loại và loại sọc nhiều nhất có thể, nó mang lại hiệu quả tốt hơn chất lượng của mã nơi tôi biết chính xác loại nào được chấp nhận và nó sẽ tránh được các lỗi không mong muốn

****

* Vì vậy hãy nói về việc định **kiểu** - Hãy để tôi làm rõ những điều này và hãy xác định **một biến** ở đây x bằng 5 - Hãy đặt **biến** đó vào **chuỗi**
* Hãy thực hiện **var\_dump x** và chúng ta nhận được giá trị 5 trong **kiểu dữ liệu chuỗi** , bạn thực sự có thể buộc đây là **một kiểu số nguyên** và bạn có thể chuyển bất cứ thứ gì sang **kiểu dữ liệu** bằng cách sử dụng tên của **kiểu dữ liệu** trong dấu ngoặc đơn

****

* Vì vậy nếu bạn muốn **chuyển chuỗ**i này thành **một số nguyên**, bạn mở dấu ngoặc đơn, bạn nhập hoặc **số nguyên** và bạn đóng dấu ngoặc đơn để bây giờ chuyển chuỗi 5 thành số nguyên 5 .

****