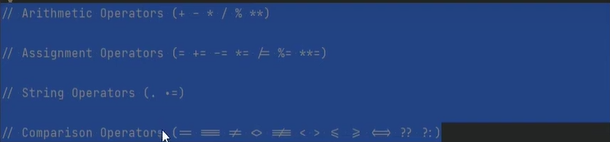
**PHP Operator Precedence & Associativity - Full PHP 8 Tutorial**

Trong video trước chúng ta đã thảo luận về các toán tử này và có các toán tử so sánh và chuỗi gán số học.



Bây giờ hãy chuyển sang toán tử kiểm soát lỗi và đó là dấu cộng ngay bất kỳ toán tử điều khiển nào làm là nếu bạn thêm nó vào một biểu thức thì nó sẽ chỉ đơn giản là ngăn chặn bất kỳ lỗi nào từ biểu thức đó và các lỗi mà nó ngăn chặn chủ yếu phụ thuộc vào cách bạn định cấu hình xử lý lỗi trong php.

Ví dụ: nếu tôi có x bằng

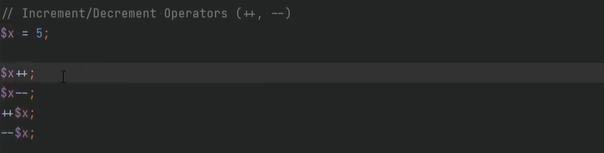
file.txt:

C:\Users\ADMIN\OneDrive\Hình ảnh\Ảnh chụp màn hình\Ảnh chụp màn hình 2023-09-07 211517.png

Và đừng lo lắng về chức năng tệp ở đây thì về cơ bản nó chỉ đọc tệpnhưng trong trường hợp này tệp này không tồn tại vì vậy nếu chúng tôi làm mới trang,chúng tôi sẽ nhận được cảnh báo rằng tệp hiện không tồn tại. Nếu bạn muốn khắc phục lỗi này, bạn có thể thêm dấu thêm ngay phía trước và sau đó nếu bạn làm mới trang, lỗi đã biến mất, tôi hoàn toàn không khuyên bạn nên sử dụng toán tử này trừ khi bạn thực sự có trường hợp sử dụng tốt cho nó. Trong một số trường hợp, có thể bạn muốn chặn một số cuộc gọi api hoặc thứ gì đó nhưng tôi vẫn không khuyên bạn nên sử dụng toán tử này. Tôi chỉ muốn chỉ cho bạn toán tử này và chỉ cho bạn cách nó hoạt động trong trường hợp bạn gặp nó trong một số cơ sở mã khác nhưng cá nhân tôi sẽ không sử dụng adsine vì nó chỉ đơn giản là làm im lặng ứng dụng của bạn nó không giải quyết được bất kỳ vấn đề nào nó không giải quyết được vấn đề của bạn và bạn có thể thậm chí không biết rằng ứng dụng của bạn có những lỗi này vì bạn đã khắc phục nó. Vậy nên tóm lại là đừng sử dụng toán tử này, hãy cố gắng xây dựng cách xử lý lỗi tốt hơn xung quanh ứng dụng của bạn và chúng tôi sẽ đề cập đến việc xử lý lỗi chi tiết hơn trong video riêng biệt, vì vậy hãy cùng xem chuyển sang toán tử tăng và giảm, bạn có thể sử dụng toán tử tăng để tăng giá trị lên một hoặc sử dụng toán tử giảm để giảm giá trị đi một. Có hai loại toán tử tăng và giảm

và đây là các toán tử giảm trước và sau tăng,

ví dụ như bạn có thể có x bằng 5 và sau đó bạn có thể thực hiện tăng bài và sau đó bạn có thể thực hiện giảm bài này rồi bạn có thể thực hiện tăng trước và sau đó giảm trước như thế này



Vậy nên việc giảm bài đăng ở trên nó làm gì đầu tiên trả về giá trị và sau đó thực hiện mức tăng tương tự cho lần tăng sau, trước tiên nó trả về giá trị và sau đó giảm giá trị đó trong trường hợp tăng trước và

giảm trước, trước tiên nó tăng và giảm rồi trả về giá trị. vì vậy

để áp dụng điều này vào thực tế, hãy cùng thực hiện thực hiện echo x ++

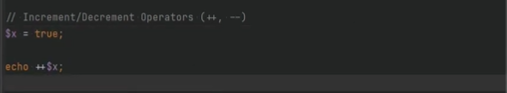
C:\Users\ADMIN\OneDrive\Hình ảnh\Ảnh chụp màn hình\Ảnh chụp màn hình 2023-09-07 214053.png

và chúng ta nhận được 5 và lý do là vì trước tiên nó trả về giá trị và sau đó tăng dần. Nếu chúng ta thực hiện echo x ngay sau khi chúng ta nhận được 6.

Tuy nhiên nếu chúng ta thực hiện tăng trước



Thì chúng ta sẽ nhận được sáu ngay lập tức vì vậy nếu chúng ta làm mới, chúng ta sẽ nhận được sáu và sáu ngay sau đó, điều tương tự cũng áp dụng cho toán tử giảm dần. Lưu ý nhanh ở đây là toán tử tăng và giảm chỉ ảnh hưởng đến số và chuỗi, các tài nguyên và đối tượng boolean không bị ảnh hưởng. Nếu chúng ta có x bằng true và chúng ta đã cộng cộng với x thì chúng ta làm mới chúng

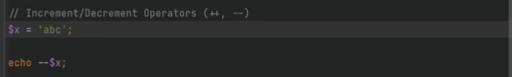


ta sẽ nhận được một giá trị nên nó sẽ không tăng true thành giá trị khác thì việc giảm giá trị null không có tác dụng nhưng việc tăng null sẽ dẫn đến một giá trị.

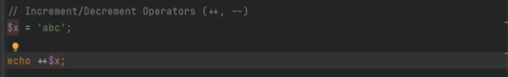
ví dụ như nếu chúng ta không làm vậy và sau đó cộng cộng x



chúng ta sẽ nhận được 1 nếu chúng ta giảm dần thì nó không ảnh hưởng gì đến nó nên nó sẽ không trả về gì cả. Giảm dần cũng không ảnh hưởng đến các giá trị chuỗi nhưng việc tăng dần có ảnh hưởng đến các giá trị chuỗi. ví dụ như nếu chúng ta có abc ở đây và chúng ta đã giảm



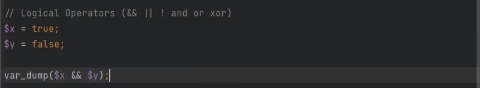
thì nó sẽ không ảnh hưởng gì nhưng nếu chúng ta ntăng



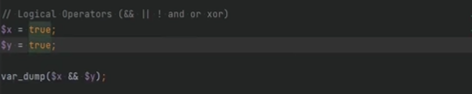
thì nó sẽ chỉ tăng các ký tự nên nếu làm mới, chúng ta sẽ nhận được abd

thay vì abc vì vậy bây giờ hãy chuyển sang các toán tử logic,

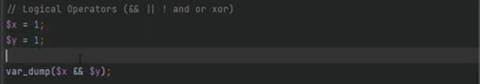
các toán tử logic về cơ bản cho phép bạn kết hợp nhiều điều kiện với nhau để bạn có thể sử dụng các toán tử này để kết hợp và tạo một câu lệnh điều kiện lớn hơn về cơ bản, hãy xem xét từng điều kiện một, chúng ta có toán tử cuối và nó chỉ đánh giá là đúng nếu cả hai toán hạng đều đánh giá là đúng. Chẳng hạn như nếu chúng tôi có x bằng true và sau đó y bằng false và sau đó chúng tôi đã kết xuất thành x và y



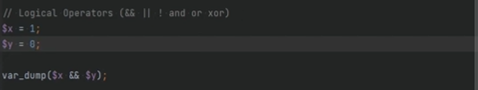
Điều này sẽ chỉ được đánh giá là đúng nếu cả x và y đều đánh giá là đúng trong trường hợp này y đánh giá là sai và do đó thao tác này ở đây sẽ đánh giá thành sai nếu chúng ta làm mới trang, chúng ta sẽ nhận được sai nếu chúng ta thay đổi y thành đúng



Bây giờ chúng ta sẽ thành đúng, chỉ cần lưu ý ở đây x và y không cần phải là giá trị boolean, có thể là số nguyên hoặc bất kỳ loại dữ liệu nào khác như bạn biết php thực hiện chuyển đổi loại cho bạn để chúng tôi có thể đặt giá trị này thành 1. Vậy chúng tôi cũng có thể đặt giá trị này thành 1 và giá trị này cũng sẽ trả về đúng



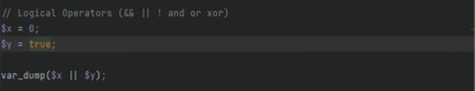
Chúng tôi có thể đặt giá trị này thành 0 và sau đó nó sẽ sai



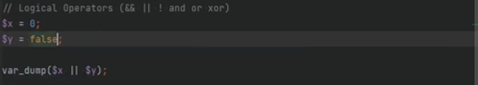
toán tử tiếp theo là hoặc và nó đánh giá thành đúng nếu x hoặc y đánh giá là đúng vì vậy nếu chúng ta thay đổi điều này thành hoặc

C:\Users\ADMIN\OneDrive\Hình ảnh\Ảnh chụp màn hình\Ảnh chụp màn hình 2023-09-07 221146.png

thao tác này sẽ đánh giá thành đúng ngay bây giờ vì x đánh giá là đúng và nếu chúng ta thay đổi xung quanh và làm theo cách này



thìđiều này sẽ đánh giá lại thành đúng nhưng nếu chúng ta thay đổi cái này thành sai

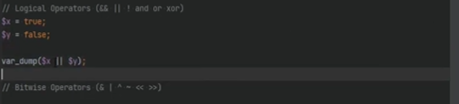


Bây giờ nó sẽ đánh giá thành sai cái thứ ba là toán tử một ngôi và nó chỉ là một phủ định nên nếu anh ta không làm x hoặc y thì điều nó sẽ làm là nó sẽ phủ nhận bất cứ giá trị nào của x trong trường hợp này x đánh giá thành false vì nó bằng 0 và khi đó phép phủ định sẽ làm cho nó đúng và vì biểu thức này ở đây đánh giá là đúng nên toàn bộ biểu thức sẽ đánh giá là đúng vì chúng ta đang sử dụng toán tử hoặc nên nếu chúng ta làm mới trang, chúng ta sẽ nhận được true php cũng có một giây biến thể của các toán tử này và đây là các từ khóa “và” “hoặc” và “sắp xếp” mặc dù phần cuối này. Và phần cuối này thực hiện cùng một điều về mặt logic, chúng thực sự khác nhau trong php và sự khác biệt nằm ở mức độ ưu tiên của chúng và chúng ta sẽ đề cập đến mức độ ưu tiên trong một phần riêng biệt video vì đây là một chủ đề rất quan trọng nên hãy để tôi cho bạn xem một ví dụ. Nếu chúng ta có x bằng true và y bằng false và sau đó chúng ta

đặt z bằng x và y và sau đó chúng ta có kết xuất z thì chúng ta nhận được sai và điều đó được mong đợi là đúng vì một trong số chúng là sai tuy nhiên nếu bạn sử dụng từ khóa and như này



bạn cũng sẽ mong đợi kết quả cũng sai nhưng nếu bạn làm mới trang thì chúng tôi nhận được giá trị đúng và lý do cho điều đó là do mức độ ưu tiên của toán tử gán có độ ưu tiên cao hơn hơn toán tử cuối ở đây và do đó cái này được đánh giá trước, sau đó cái này được đánh giá sau và nó bị loại bỏ. Vì vậy đây là lý do tại sao bạn nên cẩn thận nếu quyết định sử dụng những từ khóa này. php thực hiện một thứ gọi là đoản mạch khi nói đến toán tử logic, hãy giả sử rằng chúng ta có x bằng true và y bằng false và sau đó chúng ta làm x hoặc y,



chúng ta sẽ nhận được true nhưng biểu thức y ở đây thực sự không bao giờ được đánh giá vì nó bị đoản mạch để logic hoặc toán tử chỉ đánh giá thành

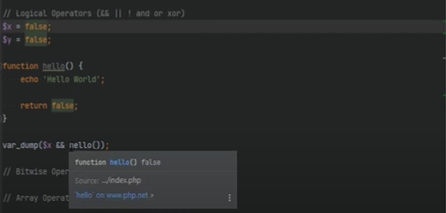
true một trong hai các toán hạng cần đánh giá là đúng và trong trường hợp này toán hạng đầu tiên được đánh giá là đúng và do đócác toán hạng còn lại không cần đánh giá và php làm ngắn mạch chúng và không đánh giá nó nên câu trả lời cuối cùng là đúng mà không cần thực thi y cho bạn thấy điều này trong một ví dụ tốt hơn, tôi sẽ tạo một hàm nhanh ở đây và đừng lo lắng về các hàm mà chúng tôi sẽ đề cập đến các hàm trong một video riêng. Hãy lưu ý với tôi ở đây giả sử rằng chúng ta có một hàm tên là

xin chào



Và tất cả những gì nó làm là nó vang vọng ra “hello world” và trả về sai nếu chúng ta sử dụng hello ở đây bạn sẽ nghĩ rằng “hello world” sẽ được in trên màn hình và nếu chúng ta làm mới trang thì nó không hiển thị

và đó là vì “hello” thực sự sẽ không bao giờ chạy nếu chúng ta thay đổi giá trị này thành toán tử kết thúc và chúng tôilàm mới trang chúng tôi thấy “hello world” và false là giá trị vì để logic và toán tử đánh giá thành true cả hai toán hạng cần đánh giá thành true và vì opera đầu tiên và giá trị quá đúng nên anh ấy cũng cần kiểm tra các toán hạng khác nếu bây giờ chúng ta thay đổi x bằng false



Một lần nữa php sẽ đoản mạch cái này và hello sẽ không bao giờ chạy vì x đánh giá là sai và logic và toán tử không cần kiểm tra bất kỳ toán hạng nào khác vì toàn bộ biểu thức đã đánh giá thành sai nếu chúng tôi làm mới, chúng tôi nhận được sai, nếu bạn làm điều gì đó tương tự hoặc đúng thì xin chào sẽ được thực thi trong trường hợp này và câu trả lời là không nếu chúng tôi làm mới trang, chúng tôi nhận được giá trị đúng, điều đó có nghĩa là nó thực sự đánh giá phần này ở đây nhưng “xin chào”

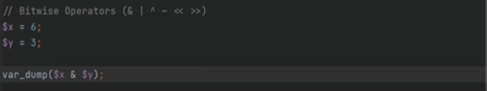
vẫn không bao giờ được thực thi và lý do cho điều đó là mức độ ưu tiên và

tính kết hợp của toán tử và chúng ta sẽ nói chi tiết hơn về điều đó trong video tiếp theo nhưng về cơ bản điều xảy ra ở đây là toán tử cuối có

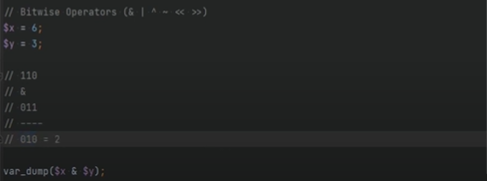
mức độ ưu tiên cao hơn toán tử hoặc và do đó nó được nhóm lại cùng nhau và được thực thi trước nên giá trị của biểu thức này là sai và vì điều này trả về sai nên toán tử hoặc cần kiểm tra các toán hạng khác và do đó, giá trị đúng được đánh giá và kết quả cuối cùng của biểu thức là đúng.

Được rồi, hãy nói về các toán tử bitwise các toán tử được sử dụng để

thực hiện các phép toán ở mức bit, hãy nghĩ về các bit khi tắt các công tắc, nó là nhị phân phải nên nó chỉ là các số 0 và các số bạn có thể sử dụng các toán tử sau để lật các công tắc đó, bạn có thể lật 0 thành một và một thành 0 hoặc dịch chuyển các bit sang trái và để có được kết quả mong muốn, toán tử đầu tiên là toán tử kết thúc và nó sẽ trả về các bit được đặt trong cả x và y, ví dụ: nếu chúng ta có x bằng sáu và y bằng ba thì chúng ta đã thực hiện var dump x y



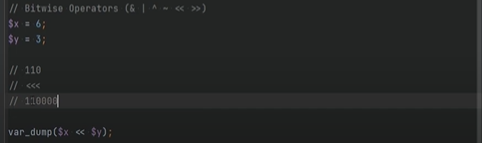
và chúng ta làm mới chúng ta đang có hai và hãy phân tích những gì chúng ta sắp đạt được sao cho biểu diễn nhị phân của số sáu là 110 và biểu diễn nhị phân của ba là 011 vì vậy chúng ta đang thực hiện phép toán kết thúc ở đây và phép toán về cơ bản trả về một nếu cả hai đều bằng một và sau đó nếu chúng ta thực hiện phép tính này theo cách thủ công, chúng ta sẽ thấy rằng một và 0 trả về 0 một và một trả về 1 và 0 và một trả về 0 và cái này ở đây bằng 2



Và đó là những gì chúng ta sắp đạt được ở đây toán tử kia đang hoặc sẽ trả về các bit được đặt theox hoặc y vì vậy nếu chúng ta thay đổi giá trị này thành hoặc và chúng ta thay đổi giá trị này thành hoặc hãy xem chúng ta sẽ nhận được gì vì vậy 1 hoặc 0 sẽ trả về 1 1 hoặc 1 sẽ trả về 1 và hoặc 1 sẽ trả về 1 và số này trong số thập phân sẽ là 1+2+4=7 vì vậy nếu chúng ta làm mới trang, chúng ta sẽ có bảy, hãy chuyển sang toán tử thứ ba là toán tử zoro vì vậy hãy đổi nó thành zor và zor chỉ là một or hoạt động độc quyền về cơ bản sẽ trả về các bit được đặt trong x hoặc y

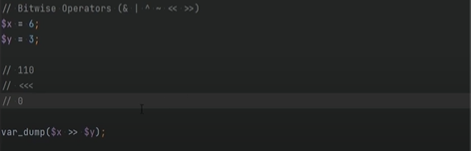
nhưng không được đặt trong cả hai. Vì vậy, ví dụ trong trường hợp này 100 sẽ trả về 1 một lần hoặc 1 sẽ trả về 0 vì nó được đặt trong cả hai và sau đó cái này cũng sẽ trả về 1 và câu trả lời cuối cùng sẽ là một cộng không cộng bốn năm và nếu chúng ta thay đổi cái này thành zor và làm mới trang thì chúng ta nhận được năm cái tiếp theo là toán tử phủ định, toán tử phủ định chỉ cần lật các bit nên nếu chúng ta không x và y thì điều gì sẽ xảy ra là chúng ta sẽ lật các bit của 6 để trả về 0 0 1 thì chúng ta sẽ thực hiện thao tác kết thúc và điều này sẽ trả về 0 0 một cũng bởi vì một và một là một và mọi thứ khác sẽ bằng 0, câu trả lời cuối cùng sẽ là một nên nếu làm mới trang, chúng ta sẽ nhận được một sau đó chúng ta có các toán tử này ở đây, toán tử này về cơ bản sẽ dịch chuyển các bit sang trái và cái này sẽ thay đổi tốc độ sang bên phải và ý tôi khi dịch chuyển các bit là mỗi khi một bit được dịch chuyển về cơ bản nó sẽ nhân với hai hoặc chia cho hai khi bạn dịch cơ số sang trái nghĩa là bạn đang nhân nó với hai nhưng khi bạn dịch chuyển cơ số sang trái thì bạn đang nhân nó với hai dịch chuyển các bit sang phải mỗi ca sẽ chia nó cho hai. Vì vậy hãy để tôi chỉ cho bạn một ví dụ để chúng ta có sáu và ba giả sử rằng chúng ta muốn dịch sáu cho ba chúng ta sẽ thực hiện x shift y và hãy thực hiện việc này một cách thủ công để chúng ta thực hiện thao tác này ở đây và hãy xóa nó đi để chúng ta sẽ dịch chuyển ba lần và điều đó có nghĩa là chúng ta sẽ thêm các số 0 vào đây nên chúng ta sẽ có một một 0 và sau đó là 0 0 vì chúng ta đang dịch chuyển ba lần và bây giờ vì chúng ta đã dịch chuyển nó ba lần nên mỗi lần chúng ta nhân nó với hai nên chúng ta có 1 2 4

8 16 16 cộng 32

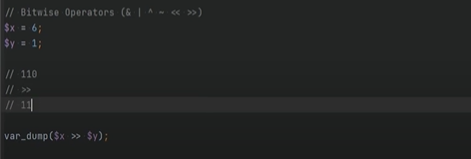


và câu trả lời cuối cùng sẽ là 48 và chúng ta sẽ nhận được 48 nếu chúng ta thực sự nhân sáu với hai ba lần nếu chúng ta làm mới trang, chúng ta sẽ nhận được 48. Nếu chúng ta dịch chuyển nó theo cách khác thay vì nối thêm các số 0 thì về cơ bản chúng ta đang loại bỏ các bit nên chúng ta sẽ loại bỏ cái này cái này và cái này và về cơ bản chúng ta chỉ còn lại

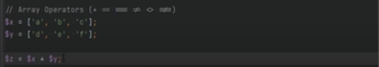
0



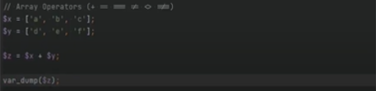
vì vậy nếu chúng ta làm mới, chúng ta sẽ nhận được số 0, hãy dịch chuyển một số nhỏ hơn vì vậy hãy dịch chuyển một số chẳng hạn tôi có một lỗi đánh máy ở đây, đây sẽ chỉ là hai và hãy dịch sáu đi một để chúng ta sẽ có một một không và sau đó chúng ta sẽ loại bỏ kết quả cuối cùng



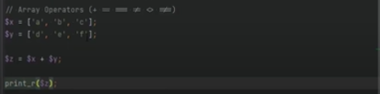
và chúng ta chỉ còn lại một cộng hai nên câu trả lời cuối cùng sẽ là ba và bạn có thể thực hiện phép tính đó bằng cách chia sáu cho hai và đó cũng là ba vì vậy nếu làm mới trang, chúng ta chỉ nhận được ba một lưu ý nhanh về các toán tử theo bit là cả hai toán hạng trong trường hợp này x và y đều được chuyển đổi thành số nguyên và sau đó các phép toán theo bit được áp dụng cho các số nguyên trừ khi cả x và y đều là các chuỗi trong đó các phép toán sẽ được thực hiện trên các giá trị ascii của các ký tự tạo nên các chuỗi đó nên bạn có thể hỏi trường hợp sử dụng của toán tử bitwise là gì và tôi sẽ nói với bạn rằng có khá nhiều trường hợp sử dụng cho chúng và bạn sẽ thấy chúng được sử dụng trong tệp cấu hình của php cũng như trong các video sau nhưng Một số trường hợp sử dụng chẳng hạn như bạn có thể sử dụng nó để mã hóa, bạn có thể sử dụng nó để lưu trữ một số cờ dưới dạng bit, thậm chí bạn có thể sử dụng nó để lưu trữ một số quyền chẳng hạn thay vì có nhiều bảng nơi bạn lưu trữ các quyền của vai trò và sau đó có một loạt các quyền tham gia nếu bạn có ứng dụng vừa và nhỏ, bạn thực sự có thể thoát khỏi bằng cách lưu trữ các quyền trong bit, hãy xem lại các toán tử mảng vì một số toán tử khi được sử dụng với mảng hoạt động hơi khác một chút vì vậy hãy xác định hai biến x bằng a b và c và sau đó y bằng d e và f nếu chúng ta làm z bằng x cộng y



thì điều gì sẽ xảy ra là toán tử cộng này sẽ chỉ tính hợp của hai mảng và phép hợp chỉ có nghĩa là lấy tất cả các phần tử từ biến y và nối nó vào biến x nếu chúng không tồn tại ở cùng một chỉ mục hoặc cùng một khóa nên trong trường hợp này cả ba phần tử ở đây đều có cùng chỉ mục là các phần tử này nên khi chúng ta thực hiện kết hợp x sẽ thực sự không thay đổi, nó sẽ chỉ sử dụng các phần tử từ biến x vì vậy nếu chúng ta thực hiện vardump z



thì chúng ta sẽ nhận được a b và c giống nhau, hãy đổi vardom thành print r để nó in tốt hơn một chút



và chúng ta sẽ thấy abc tuy nhiên nếu chúng ta thêm một vài phần tử nữa vào đây nhưng và h nếu làm mới bây giờ chúng ta sẽ thấy g và h được thêm vào mảng ban đầu nếu chúng ta thay đổi mảng này thành mảng kết hợp sao cho a bằng 1 b bằng 2 và c bằng 3 và hãy thay đổi những giá trị này ở đây



Ngoài ra, khi tính toán kết hợp trong trường hợp này, các khóa là duy nhất và các khóa không khớp nhau, do đó mọi thứ sẽ được thêm vào biến x, vì vậy nếu chúng ta làm mới trang thì bây giờ chúng ta sẽ nhận được tất cả các phần tử từ cả hai mảng nếu. Tuy nhiên một trong số chúng có cùng một khóa thì nó sẽ không ghi đè lên khóa trong biến x mà chỉ sử dụng giá trị từ biến x để làm mới giá trị cho khóa a là một chứ không phải bốn, khóa tiếp theo là so sánh và nghiêm ngặt các toán tử so sánh và cách so sánh xảy ra với mảng là nó sẽ trả về true nếu cả x và y có cùng cặp giá trị khóa, vì vậy, ví dụ trong trường hợp này

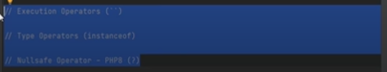


nếu chúng ta làm x bằng y thì z sẽ sai vì hai mảng không có cùng các cặp giá trị khóa ngay cả khi chúng có cùng số phần tử thì nó vẫn sai vì các khóa không khớp ngay cả khi chúng ta có các khóa khớp nhưng các giá trị khác nhau thì nó vẫn sẽ sai. Nó sẽ chỉ đúng nếu cả khóa và giá trị của mảng đều khớp nhau nên nếu chúng ta thay đổi giá trị này thành một hai và ba thì nó sẽ đúng ngay cả khi chúng ta thay đổi một trong số chúng thành một chuỗi và chúng ta đã thực hiện phép so sánh lỏng lẻo thì nó vẫn đúng vì chúng ta đang thực hiện một cách lỏng lẻo. Tuy nhiên, so sánh nếu bạn thực hiện so sánh nghiêm ngặt thì điều này cũng sẽ kiểm tra kiểu dữ liệu của các giá trị.

  
Vì vậy trong trường hợp này, điều này sẽ trả về sai vì các kiểu dữ liệu không khớp nên về cơ bản, so sánh nghiêm ngặt sẽ chỉ trả về true nếu cặp giá trị khóa giống nhau cũng như các kiểu dữ liệu của chúng và chúng cũng theo cùng một thứ tự. Vì vậy nếu chúng ta thay thế thứ tự ở đây và chúng ta di chuyển b sang b ở phía bên kia và chúng ta thay đổi nó thành cùng một kiểu dữ liệu với một so sánh nghiêm ngặt



thì điều này sẽ trả về sai nhưng nếu chúng ta thực hiện so sánh lỏng lẻo thì điều này sẽ trả về true ngay cả khi chúng không đúng thứ tự và logic tương tự áp dụng cho kiểm tra bất đẳng thức và kiểm tra bất đẳng thức nghiêm ngặt và toán tử này ở đây chỉ là toán tử thay thế để kiểm tra bất đẳng thức. Vì vậy toán tử tiếp theo chúng ta có các toán tử thực thi các toán tử kiểu và toán tử lưu null



nhưng đừng lo lắng về ba toán tử này vì bây giờ các toán tử lưu kiểu và null sẽ được đề cập đến trong phần thứ hai của khóa học này khi chúng ta đến các lớp và đối tượng vì nó không nên che nó ngay bây giờ khi chúng ta chưa che các đối tượng, toán tử thực thi có thể là thứ bạn sẽ không bao giờ sử dụng. Nó cho phép bạn thực thi nội dung của backticks dưới dạng lệnh shell chỉ khi shell exec được bật nên đừng lo lắng về cái này cũng vì bạn có thể sẽ không cần nó trừ khi bạn cần thực thi một số lệnh shell từ php. Vì vậy đây là cái dành cho video này, cảm ơn bạn rất nhiều vì đã xem và trong video tiếp theo, chúng ta sẽ nói về một chủ đề rất quan trọng đó là về quyền ưu tiên của toán tử.