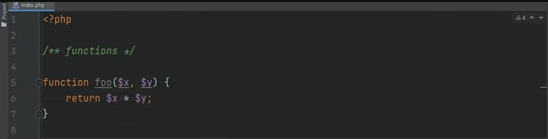
1.22: PHP Function Parameters - Named Arguments - Variadic Functions & Unpacking - Full PHP 8 Tutorial (Tham số hàm PHP - Đối số được đặt tên - Hàm biến đổi & Giải nén)

1. Thông số chức năng:

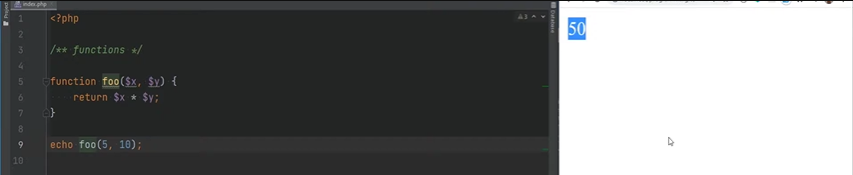
* Trong video trước bạn sẽ tìm hiểu cơ bản về các chức năng cách tạo chức năng làm thế nào để trả về giá trị các kiểu trả về, v.v.
* Trong video này chúng ta sẽ nói về sự chấp nhận đối số các giá trị mặc định giải nén các loại đối số và hơn thế nữa bạn có thể xác định

1. Xác định tham số hàm

* Các tham số của hàm trong dấu ngoặc đơn của định nghĩa hàm
* Vì vậy, ví dụ giả sử chúng ta có một hàm được gọi là foo để chúng ta có thể nói x dấu phẩy y dấu phẩy và nhiều tham số khi bạn cần hãy nói rằng chức năng này chỉ đơn giản là nhân x với y và trả về giá trị đó



* Khi chúng ta gọi hàm đó, chúng ta cần chuyển lập luận để chúng ta có thể nói 5 và 10 và chúng ta có thể lặp lại việc làm mới đó trang chúng tôi nhận được 50



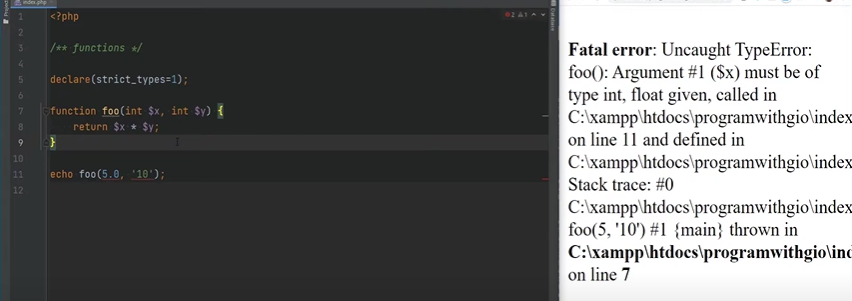
1. Tham số và đối số

* Tham số và tham số đối số là những x và y này ở đây khi chúng được xác định trong định nghĩa hàm nên chúng tôi gọi chúng là các tham số các đối số là thực tế các giá trị mà chúng ta truyền cho hàm. Những thông số đó nên trong trường hợp này x và y là các tham số và 5 và 10 là những đối số đang được thông qua giống như bạn có thể gõ gợi ý

1. Gõ gợi ý

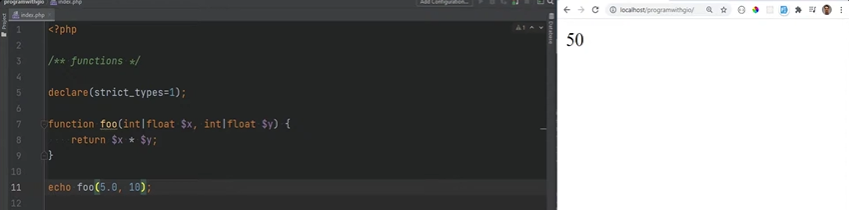
* Trả về giá trị bạn cũng có thể nhập trong các tham số để chúng ta có thể nói rằng chúng ta đang mong đợi, x là số nguyên và y là số nguyên. Bây giờ nếu chúng ta vượt qua một cái gì đó như nổi hoặc xâu chuỗi tốt

Điều này vẫn sẽ làm việc vì chúng tôi không thực thi các loại nghiêm ngặt như đã đề cập trước đây để làm rằng chúng ta có thể khai báo các loại nghiêm ngặt và bây giờchúng tôi sẽ gặp lỗi. Tôi đã đề cập đến điều này trước đây nhưng tôi sẽ nói lại lần nữa tôi cá nhân tôi thích sử dụng các loại nghiêm ngặt và tôi rất khuyến khích điều đó bởi vì khi bạn xác định rõ ràng các loại về cơ bản bạn biết chính xác những gì loại đối số dự kiến là gì và loại dự kiến của giá trị trả về là bạn muốn chấp nhận nhiều loại

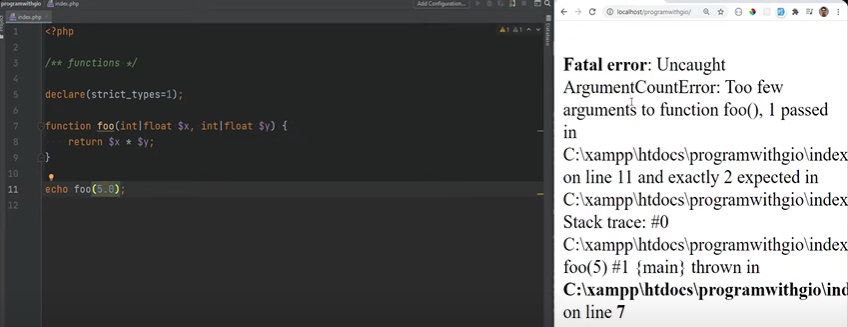


1. Các loại toán và tin

* Ở đây, chúng ta vẫn sẽ thất bại vì chúng ta đang chuyển chuỗi ở đây để có thể thay đổi lại thành số nguyên và làm mới và mọi thứ hoạt động như được mong đợi như ta đã đề cập trước đó, tôi đánh giá cao khuyên bạn nên sử dụng loại nghiêm ngặt đặc biệt khi bạn đang sử dụng các loại kết hợp vì ép buộc kiểu có thể phức tạp và dẫn đến kết quả với những giá trị không mong đợi tôi sẽ rời đi liên kết trong mô tả đến tài liệu nơi bạn có thể đọc thêm về nó cho mọi tham số bạn cần vượt qua một đối số



* Nếu không bạn sẽ nhận được một lỗi chẳng hạn như nếu chúng ta quên vượt qua thứ hai tranh luận ở đây và chúng ta làm mới trang chúng ta sẽ gặp lỗi



1. Giá trị mặc định

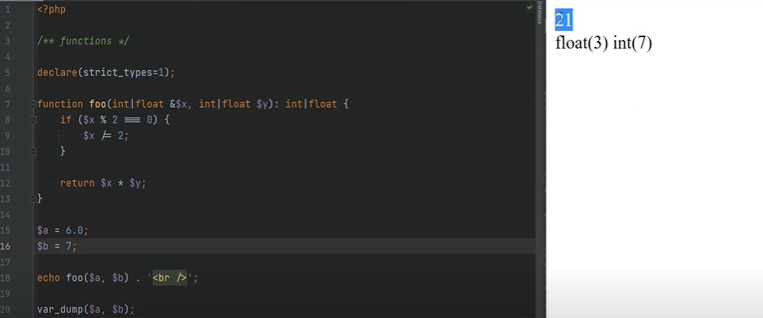
* Để đặt mặc định của các tham số để bạn có thể nói rằng bạn theo mặc định nếu không được chuyển vào có thể được đặt đến 10 chẳng hạn và bây giờ khi chúng ta làm mới trang mọi thứ vẫn hoạt động, bạn có thể gán mảng vô hướng và giá trị null như mặc định nó không thể là một lệnh gọi hàm hoặc nó không thể là một vật thể hoặc những thứ như thế nó phải là một hằng số.
* Biểu hiện một điều khác cần lưu ý là các tham số tùy chọn phải được xác định sau bất kỳ tham số không tùy chọn nào, vì vậy bạn không thể làm điều gì đó như có yêu cầu tham số và sau đó có một tùy chọn tham số và sau đó có một tham số khác được yêu cầu tham số này sẽ không hoạt động đâu, bạn phải có các thông số tùy chọn của bạn sau tất cả các thông số cần thiết

1. Truyền đối số theo giá trị so với tham chiếu

* Các đối số theo mặc định được truyền bởi giá trị và bạn có thể thay đổi điều đó và thay vào đó chỉ định chuyển bằng tham chiếu trong chức năng. Ví dụ này, foo chấp nhận hai đối số và nếu đối số đầu tiên là số chẵn thì nó chia đôi. Nếu không thì nó sẽ nhân số đầu tiên đối số với đối số thứ hai bởi mặc định khi bạn truyền đối số chúng được truyền theo giá trị nên đây là được đánh giá là 6 và 6 được chuyển theo giá trị
* Nếu bạn thay đổi x ở đây giống như chúng ta đang làm đúng vậy, nói x bằng x chia cho 2 vậy x bây giờ trở thành 3. Nó không ảnh hưởng đến biến ban đầu chúng ta thấy ở đây rằng biến a vẫn là sáu, nó có thể thêm phần trăm vào ở đây và bây giờ điều này cho thấy rằng chúng tôi chuyển đối số ở đây bằng cách tham khảo



* Vì vậy trong trường hợp này khi chúng ta đi qua a thay vì chuyển giá trị sáu chúng ta truyền tham chiếu tới biến a. Vì vậy ngay bây giờ nếu chúng tôi làm mới thì chúng tôi vẫn sẽ nhận được 21 nhưng khi bạn nhận thấy sáu ở đây đổi thành ba nên a là không lâu hơn sáu bây giờ bằng ba vì chúng ta đã sửa đổi biến trong hoạt động được rồi

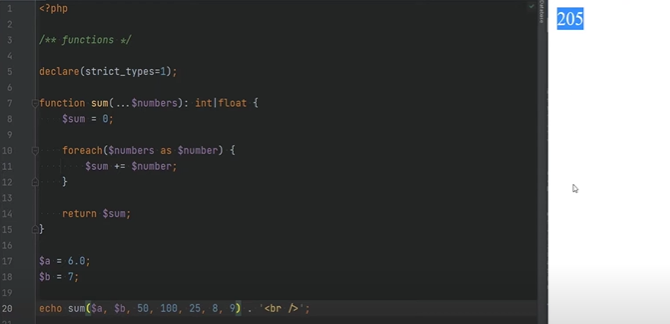


1. Hàm biến đổi - toán tử splat - nắm bắt các đối số được truyền

* Toán tử splat và định kỳ hàm số hàm tuần hoàn là chức năng chấp nhận số lượng đối số thay đổi, tôi sẽ đổi tên hàm này thành tổng và xóa chỉ báo tham chiếu ở đây và chỉ cần nói x cộng y bây giờ nếu chúng ta làm điều này thì chúng ta sẽ để có được 13.



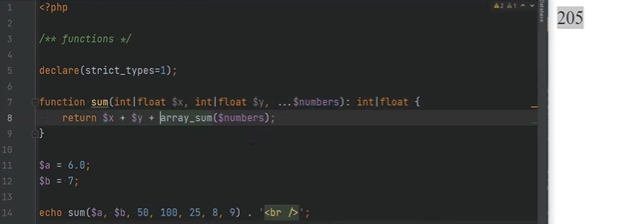
Nhưng hiện tại chúng tôi bị giới hạn với hai lập luận đúng ý tôi là chúng ta có thể vượt qua cái khác giá trị ở đây như 50. Nó không nhận được bao gồm trong tính toán, bạn có thể thêm một định nghĩa tham số khác và chấp nhận tham số thứ ba và sau đó thêm nó vào tính toán. Nhưng sau đó bạn sẽ phải làm điều đó cho mọi lý lẽ rằng bạn vượt qua, vì vậy thay vì xác định nhiều các thông số bạn chỉ có thể sử dụng biểu tượng toán tử đó là dot dot dot và sau đó tên biến và nó sẽ nắm bắt các đối số đang được chuyển đến hàm thành một mảng và số .Bây giờ sẽ là một mảng chứa tất cả những lý lẽ đó, vì vậy bây giờ bạn có thể làm tất cả các loại tính toán về điều này bởi vì nó là một mảng và bạn có thể sử dụng nhiều được xây dựng trong các hàm mảng bạn cũng có thể lặp lại trên mảng đó và chỉ tính toán tính tổng theo cách thủ công để bạn có thể tính tổng bằng đến 0 và sau đó cho mỗi số là số tổng cộng bằng số và trả về tổng hợp bây giờ chúng ta có thể chuyển nhiều đối số như chúng ta muốn và chúng ta làm mới, chúng ta nhận được giá trị chính xác



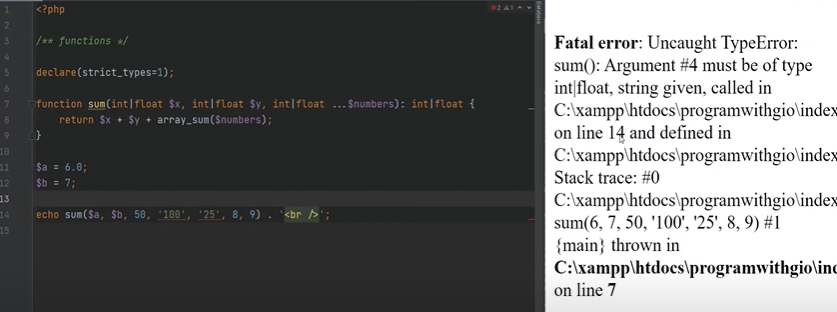
* Thay vì làm điều này bạn thực sự có thể sử dụng một mảng được xây dựng trong hàm để tính tổng giá trị ta có thể gọi số tổng của mảng và loại bỏ điều này và làm mới, ta nhận được cùng một giá trị



* Lưu ý rằng toán tử splat có thể được sử dụng sau khi sửa lỗi số lượng tham số trong trường hợp này chỉ các đối số được thông qua sau. Điều đó sẽ được thêm vào mảng có nghĩa là nó có thể định nghĩa theo cách chúng ta đã có trước int tải x và float y và chỉ sau lần chụp đó tất cả các đối số khác thành những con số mảng vậy bây giờ điều gì sẽ xảy ra là x và y đó sẽ bằng 6 và 7 và mọi thứ phần còn lại sẽ được ghi lại trong các con số mảng nên ở đây chúng ta có thể làm x cộng y cộng xóa một số số và chúng ta sẽ nhận được kết quả tương tự



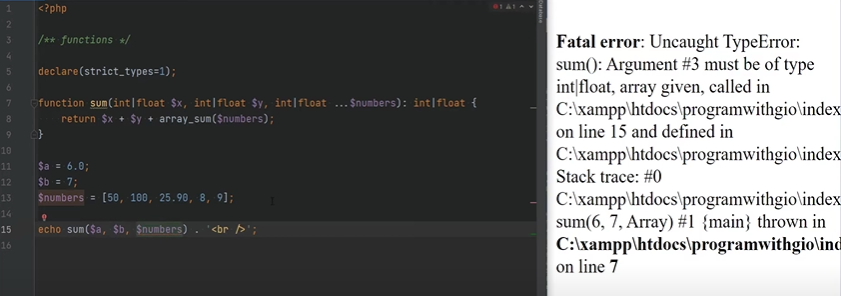
* Bây giờ trong thực tế, ứng dụng ta có thể sẽ không bao giờ làm một cái gì đó như thế này. Trong ứng dụng thực tế, bạn có thể sẽ làm một số loại xử lý bằng có lẽ những lập luận đang được thông qua ta sẽ lặp lại chúng thực hiện một số kiểm tra một số xử lý và chỉ khi đó ta mới làm được tính toán nhưng đây chỉ là để hiển thị ta biết nó hoạt động như thế nào, ta cũng có thể gõ gợi ý tất cả các lập luận đang được đã trôi qua ngay bây giờ chúng ta có thể vượt qua điều này như chuỗi và nó vẫn hoạt động, chúng ta không sẽ gặp bất kỳ lỗi nào mặc dù chúng ta đang ở chế độ nghiêm ngặt, ta có thể nhập các số ở đây cũng giống như vậy gợi ý hoặc thả nổi và bây giờ tất cả các đối số đang được vượt qua và nắm bắt bởi mảng phải là thuộc loại đó để ta có thể thấy nó gạch chân hai điều này và chúng ta sẽ gặp lỗi



* Chúng ta sẽ phải thay đổi điều này đến số nguyên hoặc số float ta có thể hỏi, ta không thể chỉ sử dụng một mảng và chấp nhận một tham số dưới dạng một mảng và chỉ cần truyền những lập luận này như những phần tử bên trong một mảng câu trả lời cho điều đó là có nhưng toán tử splat cho phép ta xác định các loại của mỗi phần tử trong một mảng ta không thể định nghĩa kiểu toán tử splat

1. Giải nén đối số bằng toán tử splat (elipsis/ba dấu chấm)

* Có thể được sử dụng để giải nén mảng thành danh sách đối số chẳng hạn nếu ta đã có những thứ này trong một mảng vì vậy giả sử chúng ta có số ở đây chúng ta không thể chuyển cái này vào đúng. Chúng ta sẽ gặp lỗi vì đây không phải là số nguyên hay float



* Bạn thực sự có thể giải nén những điều này thành những đối số riêng biệt và bạn có thể làm điều đó bằng cách sử dụng toán tử splat. Ở đây để bạn có thể thêm ba dấu chấm ở đây và điều sắp xảy ra là nó sẽ chuyển chúng thành các đối số riêng biệt vì vậy nếu chúng tôi làm mới chúng tôi sẽ nhận được kết quả tương tự



1. Đối số được đặt tên PHP8

* Đối số được đặt tên cho phép bạn vượt qua các đối số cho hàm dựa trên biến tham số tên có nghĩa là nó không cần phải theo thứ tự. Khi bạn chuyển đối số cho chức năng bạn phải vượt qua chúng theo thứ tự trong đó chúng được xác định trong trường hợp này, chúng ta đang xác định tham số x và tham số y và chúng ta ở đây kiểm tra xem x có chia hết cho y không, chia đều rồi đem trả lại. Nếu không thì chỉ cần trả lại x và ở đây, chúng ta đang chuyển các giá trị cho x và y và chúng ta sẽ nhận được kết quả chính xác



* Nhưng điều gì sẽ xảy ra nếu bạn làm sai thứ tự và bạn đặt y trước và sau đó bạn đặt x sau bạn làm mới và bạn sẽ nhận được 3, đó là giá trị của y bây giờ. chuyện gì sẽ xảy ra nếu bạn đang chuyển giá trị của y là 3 ở vị trí của tham số x và 3 thì không chia hết cho 6 và do đó bạn đang quay trở lại 3. Vì php 8 php thực sự cho phép bạn đặt tên cho lập luận của bạn để bạn có thể nói điều đó. Cái này là giá trị của tham số y và đây là giá trị cho tham số x bây giờ nếu chúng ta làm mới chúng ta sẽ nhận được kết quả đúng và thứ tự không còn là vấn đề nữa



1.23: PHP Variable Scopes - Static Variables (Phạm vi biến PHP - Biến tĩnh)

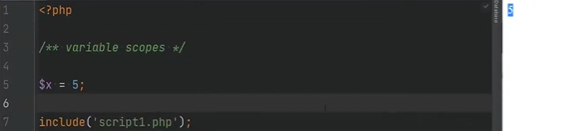
1. Phạm vi biến đổi:

- Phạm vi biến chỉ đơn giản là chỉ ra ranh giới nơi biến có thể truy cập, chúng tôi có phạm vi toàn cầu và phạm vi địa phương chỉ dành cho hầu hết các biến trong php. Có một phạm vi duy nhất và nó trải rộng trong các tập tin được bao gồm và yêu cầu chúng ta sẽ nói nhiều hơn về phạm vi khi chúng ta truy cập vào php hướng đối tượng

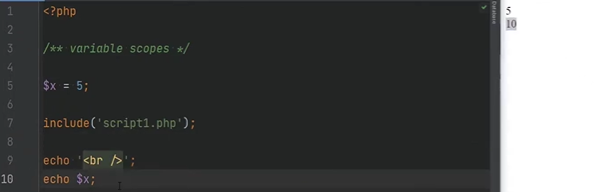
2. Phạm vi toàn cầu:

- Nó nằm trong phạm vi toàn cầu có nghĩa là nó có sẵn trong suốt tập lệnh cũng sẽ có sẵn cho các tập lệnh được bao gồm sau khi sử dụng bao gồm hoặc yêu cầu tuyên bố như vậy

- Đối với ví dụ, nếu tôi bao gồm script1 dot php biến x sẽ có sẵn trong tập tin đó và tôi sẽ có thể sử dụng nó. Vậy hãy để tôi làm nhanh việc đó để tôi tạo tập tin này và bây giờ tôi sẽ gọi x từ đây nếu chúng ta làm mới trang nó sẽ in tốt



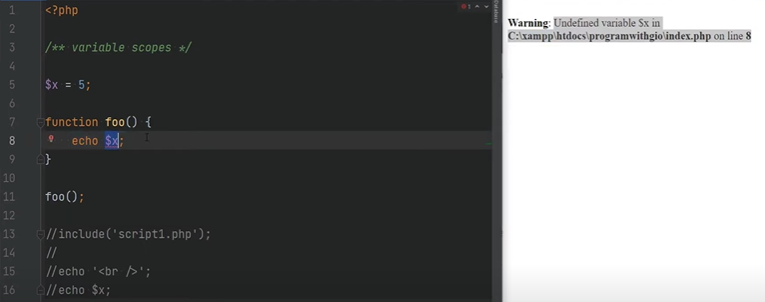
* Nếu chúng ta cố gắng thay đổi x ở đây thành 10, quay lại trang chỉ mục và cố gắng lặp lại nó ở đây khi làm mới trang chúng ta sẽ nhận được 10



3. Phạm vi địa phương:

- Chức năng các biến trong hàm đó có phạm vi địa phương và được giới hạn ở địa phương phạm vi chức năng

- Nếu xác định một chức năng đầy đủ ở đây và cố gắng lặp lại x từ đây và hãy để tôi bình luận những điều này và tôi sẽ gọi hàm qua đây và chúng tôi làm mới trang chúng ta sẽ nhận được cảnh báo không xác định biến x và đó là bởi vì mọi biến trong chức năng có phạm vi địa phương



* Cho biến x tồn tại ở đây chúng ta có ba tùy chọn. Một là chúng ta xác định biến trong ở đây đại loại như thế và bây giờ x sẽ bằng một



* Một lựa chọn khác là chúng tôi định nghĩa đó là một tham số và chúng tôi chuyển nó vào như một cuộc tranh luận như thế này và bây giờ chúng tôi làm mới trang và nó sẽ thành công

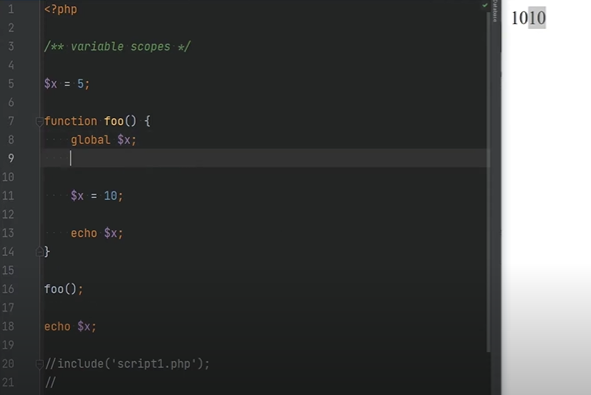


4. Sử dụng từ khóa toàn cầu để truy cập các biến trong phạm vi toàn cầu

- Tùy chọn thứ ba đang sử dụng là các từ khóa toàn cầu để truy cập các biến trong phạm vi toàn cầu vì vậy để truy cập các biến trong toàn cầu phạm vi bạn chỉ cần thêm từ khóa toàn cầu đến biến như thế này toàn cầu và x và bây giờ x sẽ có thể truy cập được trong hàm. Nếu chúng ta làm mới trang chúng ta sẽ nhận được 5.

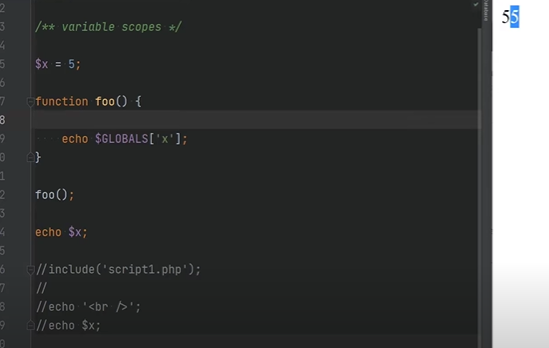


* Bạn cũng có thể thay đổi nó nếu bạn thay đổi x lên 10 bây giờ tôi sẽ in 10 nhưng cái này thực sự là một tham chiếu đến bản gốc biến vì vậy nếu chúng ta đi ở đây và lặp lại chữ x chúng ta sắp nói nhận được 10 chứ không phải 5.



5. Các biến toàn cục được lưu trữ như thế nào và siêu toàn cầu

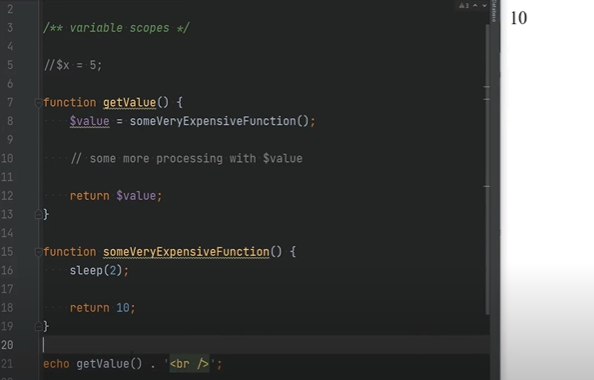
- Các biến trong một mảng kết hợp được gọi là toàn cầu trong đó khóa là tên biến và giá trị là giá trị biến vì vậy bạn có thể truy cập chúng theo cách đó thay vì sử dụng các từ khóa toàn cầu như vậy. Ví dụ chúng ta có thể ở đây thay vào đó hãy nói Globalx và lặp lại điều đó và xóa cái này khỏi đây, làm mới và như bạn có thể thấy chúng tôi nhận được 5.



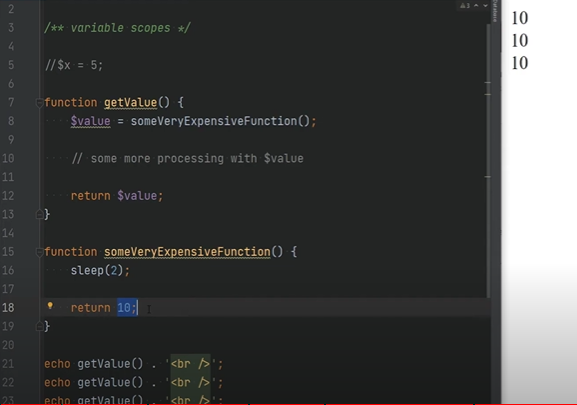
6. Biến tĩnh

- Một biến tĩnh chỉ là một biến thông thường, biến có phạm vi cục bộ, sự khác biệt là biến thông thường bị phá hủy bên ngoài phạm vi ranh giới trong khi biến tĩnh thì không bị phá hủy và nó vẫn giữ nguyên giá trị

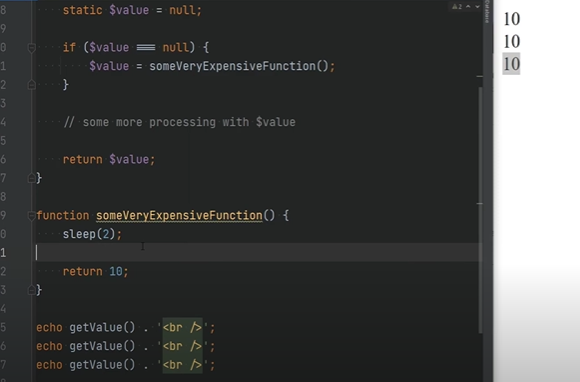
- Ví dụ tôi đã tạo một hàm ở đây được gọi là giá trị nhận được, thực hiện một số loại xử lý và nó nhận được giá trị từ một rất đắt nhưng trước khi trả về giá trị bạn có thể thực hiện một số loại xử lý với giá trị đó và sau đó trả lại vì thực tế ứng dụng bạn có thể sẽ làm một số loại xử lý với giá trị và sau đó bạn sẽ trả lại nó. Giả sử bây giờ chúng ta hãy thực hiện một chức năng rất tốn kém ở đây và chúng tôi có thể mô phỏng điều đó bằng cách sử dụng giấc ngủ chức năng để tập lệnh ngủ trong vài giây bây giờ chúng ta có thể làm tiếng vọng nhận giá trị và hãy thêm một dòng ngắt và chúng tôi làm mới trang nó sẽ đợi trong 2 giây và nó sẽ in ra số 10.



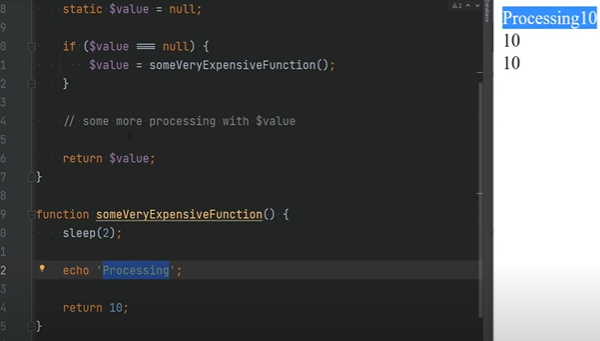
* Nếu bạn cần gọi hàm này nhiều lần vì bạn có thể muốn lấy giá trị đó từ nhiều nơi. Tôi sẽ nhân đôi nó theo cách này chỉ để giới thiệu cách nó hoạt động nhưng trong ứng dụng thực tế bạn có thể sẽ gọi cái này từ các phần khác nhau của mã của bạn nhưng ý tưởng vẫn vậy nếu tôi làm mới bây giờ thì nó sẽ diễn ra ngủ trong sáu giây vì chúng tôi gọi nó ba lần và mỗi lần nó gọi đây là một số thứ rất đắt tiền, chức năng thực hiện một số phép tính và trả về giá trị



* Bây giờ giá trị này không thực sự thay đổi đúng bởi vì chúng tôi không vượt qua bất kỳ tranh luận ở đây và chúng tôi sẽ không chuyển bất kỳ tranh luận ở đây bạn chỉ cần trả lại một số loại của dữ liệu sau khi thực hiện một số việc tốn kém xử lý bạn có thể sử dụng tĩnh các biến ở đây để lưu trữ cái này và bạn sẽ làm điều đó bằng cách sử dụng tĩnh từ khóa vì vậy bạn sẽ thêm giá trị tĩnh và đặt mặc định là null và sau đó ở đây bạn sẽ kiểm tra xem giá trị là null sau đó bạn đặt giá trị thành bất cứ giá trị nào hàm đánh giá để và nếu không thì bạn chỉ cần trả lại giá trị nên bây giờ nếu chúng ta làm mới trang thì đó là chỉ ngủ được 2 giây và nó vẫn sẽ in 10 ba lần vì chúng tôi đang trả lại nó ở đây



* Nhưng nếu tôi thêm câu lệnh echo ở đây cái này sẽ chỉ được in một lần đã đến lúc tôi làm mới nó chỉ in lần đầu tiên



* Nếu tôi xóa từ khóa tĩnh ở đây và tôi làm mới nó sắp được in 3 lần và đó là như bạn có thể thấy, các biến tĩnh có thể hữu ích để lưu trữ những thứ như thế này

