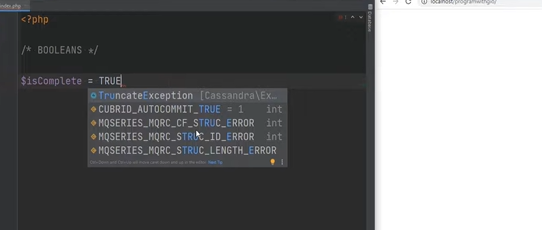
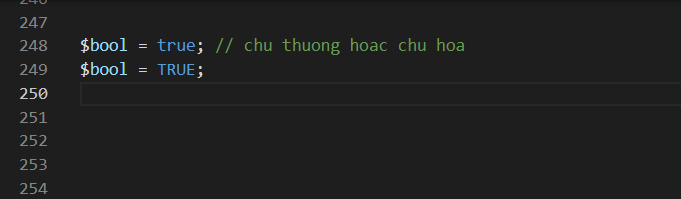
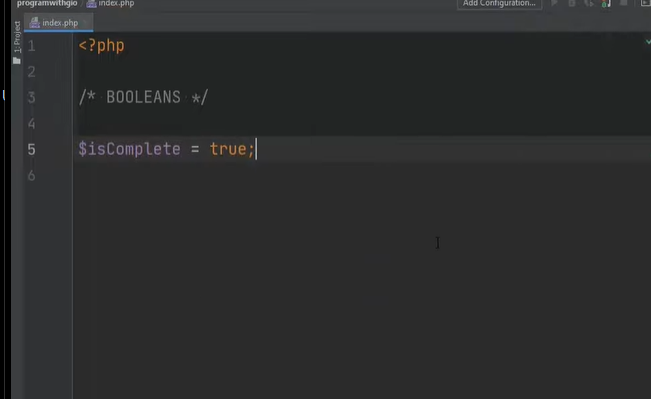
**1.5 PHP Boolean Data Type – Full PHP 8 Tutorial**

* A boolean is a simple representation of a truth value it can either be true or false
* *Boolean là một biểu diễn đơn giản của một giá trị đúng/sai, nó chỉ có thể là true hoặc false.*
* So if we have a variable called is complete we can assign either true or false to this
* *Nếu chúng ta có một biến gọi là "isComplete", chúng ta có thể gán true hoặc false cho nó.*
* And the true and false are the predefined constants that come with php
* *True và false là các hằng số đã được định nghĩa sẵn đi kèm với PHP*
* And they are also case insensitive
* *Chúng cũng không phân biệt chữ hoa chữ thường.*

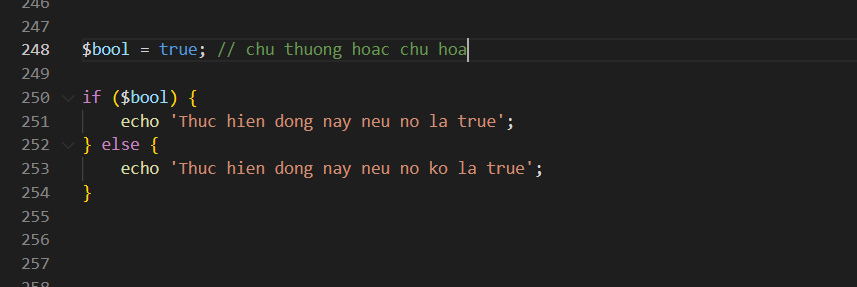
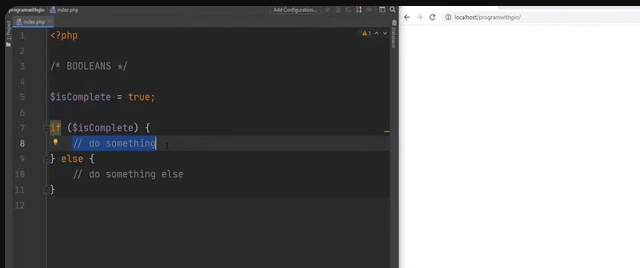




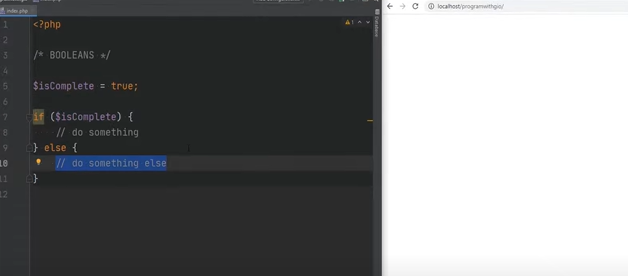
* So you can set this to true or lower case or you can use all uppercase or you can mix and match uppercase and lowercase letters
* *Vì vậy, bạn có thể đặt giá trị này là true hoặc viết thường hoặc viết hoa hoặc kết hợp chữ hoa và chữ thường*
* Though i highly recommend to just stick with one and stay consistent i personally prefer all lowercase so that's what i'll be using
* *Tuy nhiên, tôi khuyến nghị bạn nên chọn một cách viết và duy trì nhất quán. Cá nhân tôi thích viết thường nên đó là cách tôi sẽ sử dụng.*



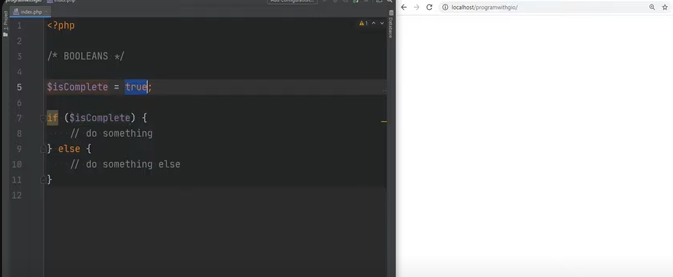
* Booleans are mostly used within control structures like loops or ethos conditionals
* *Boolean thường được sử dụng trong các cấu trúc điều khiển như vòng lặp hoặc câu điều kiện.*
* Don't worry about the control structures for now we'll talk about those in a separate video
* *Đừng lo lắng về các cấu trúc điều khiển cho đến bây giờ, chúng ta sẽ thảo luận về chúng trong một video riêng*
* But it would look something like this
* *-Nhưng nó sẽ trông giống như thế này thử với if else*



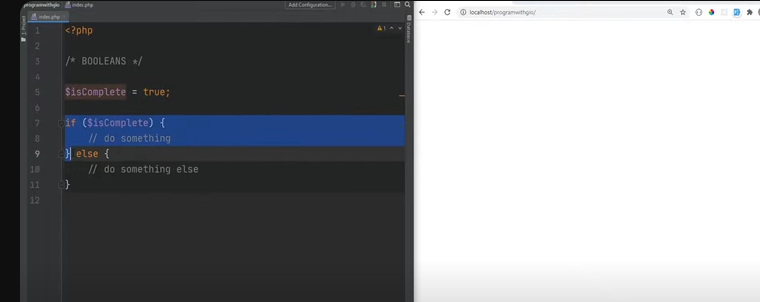
* If is complete do something otherwise do something else what this means is
* *Nếu hoàn tất làm điều gì đó , hoặc làm 1 điều gì đó khác*
* That this section right here will only evaluate if the expression evaluates to true otherwise this section will run now what
* *Phần này sẽ chỉ đánh giá biểu thức xem liệu nó có trả về giá trị true hay không. Nếu không, phần bên else sẽ được thực thi*

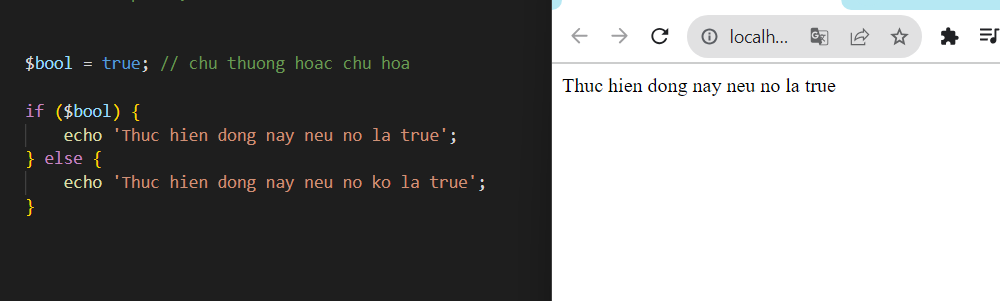


* Do i mean by an expression evaluating to True
* *Khi nói về việc một biểu thức đánh giá thành true*
* This for example is a variable and the variable is set to true so this evaluates to true
* *Ví dụ này là một biến và biến này được đặt thành true, dẫn đến việc biểu thức này đánh giá thành true*

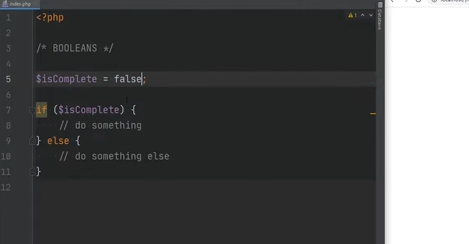


* And therefore this section will run right here
* *Do đó phần này sẽ được thực thi.*

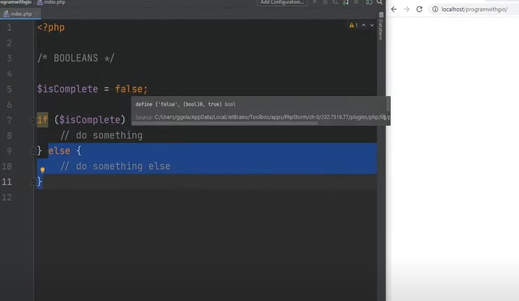


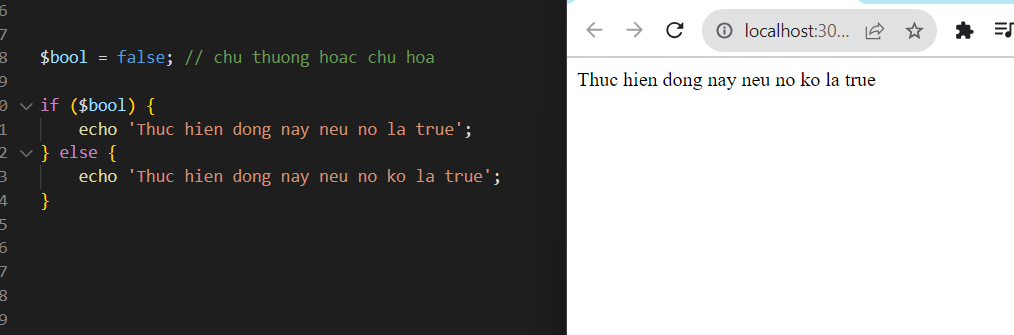


* If we were to set this to false
* *Nếu chúng ta đặt phần này là false*

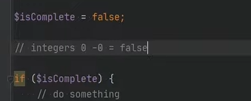


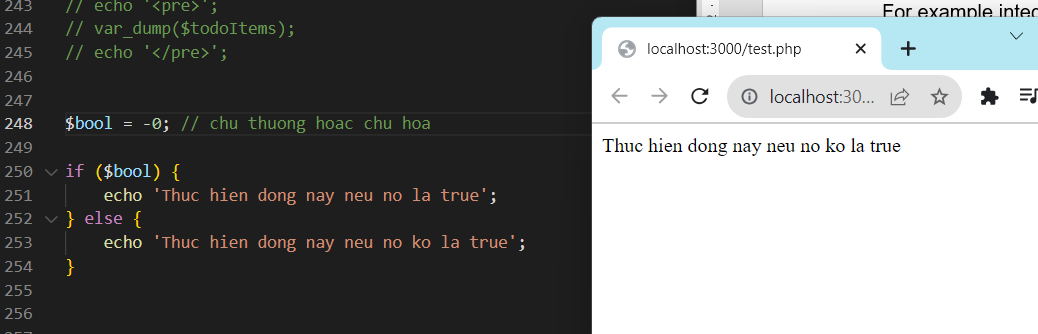
* Then this section would execute now
* *Thì phần này sẽ thực thi luôn*



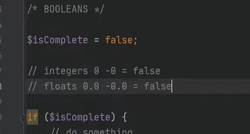


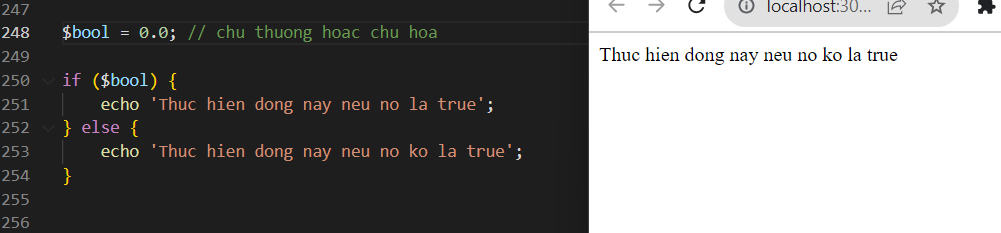
* There are other data types that could be converted to booleans on the fly by php and they could be evaluated to either true or false
* *Có các kiểu dữ liệu khác có thể được chuyển đổi thành boolean ngay lập tức bởi PHP và chúng có thể được đánh giá là true or false*
* For example integers 0 and negative 0 will always be considered as false
* *Ví dụ, số nguyên 0 và số nguyên âm 0 sẽ luôn được coi là false.*



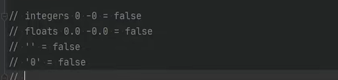


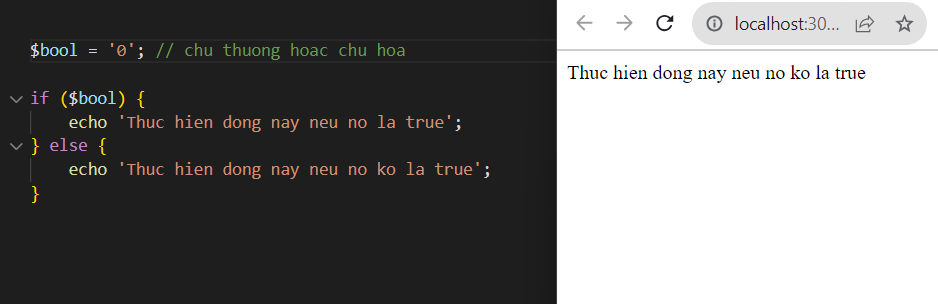
* Same applies to floats 0.0 and negative 0.0 will also evaluate to false
* *Tương tự, số thực 0.0 và số thực âm 0.0 cũng sẽ đánh giá là false.*



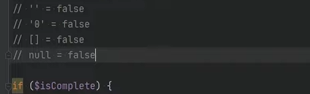


* Then we have strings if you have an empty string this will also evaluate to false or if you have a 0 that is a string within quotes like this it will also evaluate to false
* *Tiếp theo, chúng ta có chuỗi. Nếu bạn có một chuỗi rỗng, nó cũng sẽ đánh giá là false hoặc nếu bạn có một chuỗi là "0", nó cũng sẽ đánh giá là false.*

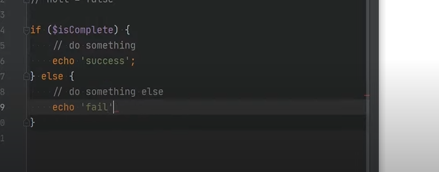




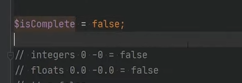
* An empty array for example like this will also evaluate to false and the special type null will also evaluate to false
* *Một mảng rỗng, ví dụ như này, cũng sẽ đánh giá là false. Và kiểu đặc biệt null cũng sẽ đánh giá là false*



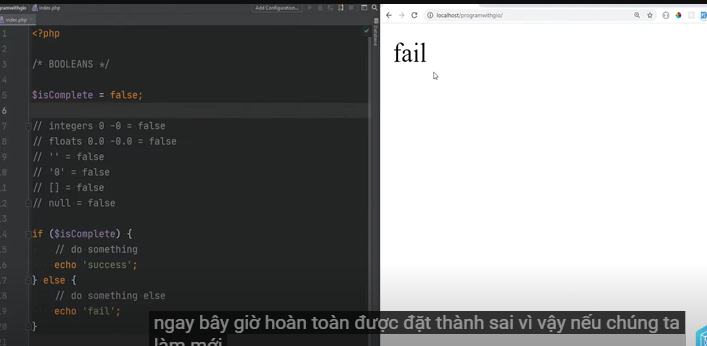
* Anything else pretty much will evaluate to true even the negative numbers that are not zero
* *Bất cứ thứ gì khác thường sẽ đánh giá là true, kể cả các số âm không phải là số không*
* So let's test this out if something evaluates to true let's print out success and otherwise let's print out fail
* *Hãy thử nghiệm điều này nếu một cái gì đó được đánh giá là true, chúng ta hãy in ra "success", nếu không, hãy in ra "fail".*

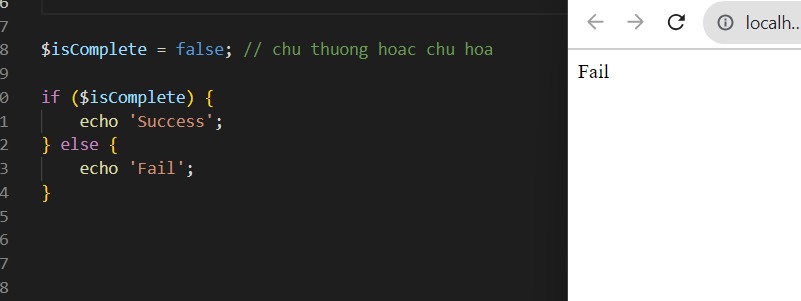


* Right now is completely set to false
* *Hiện tại biến completely được gán false*

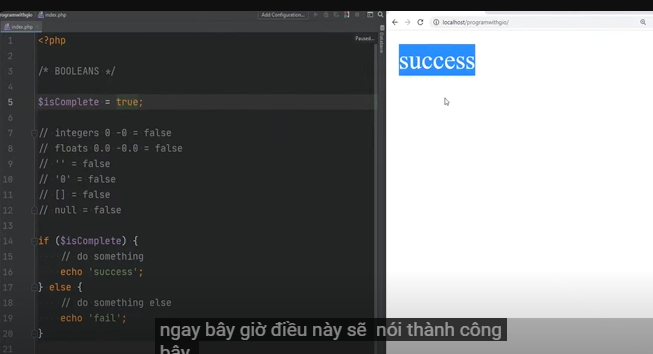


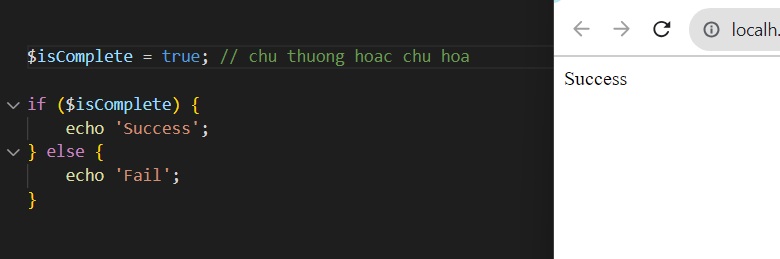
* So if we refresh the page we say fail
* *Tải lại trang thì in ra fail*



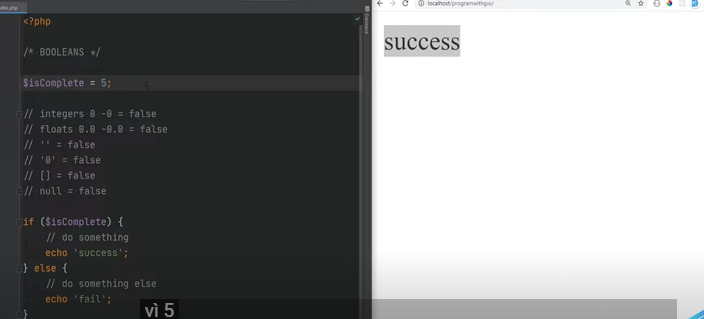


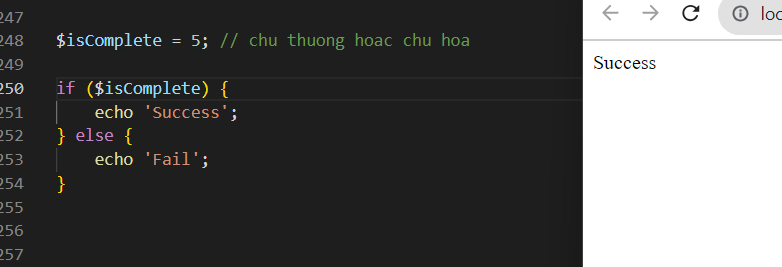
* Let's set that to true now this will say success
* *Nếu ngược lại đặt là true sẽ in ra success*



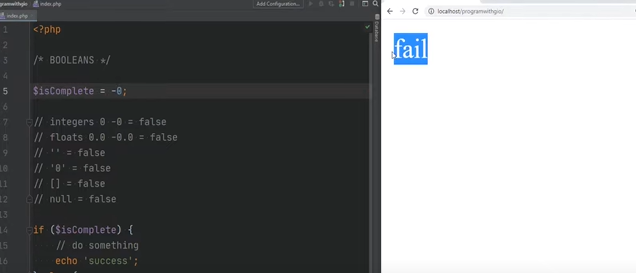


* Now let's set this to an integer number 5 and this will say success still because 5 evaluates to true
* *Nếu đặt biến này là số nguyên int 5 thì in ra success vì 5 được đánh giá là đúng (cứ số khác mấy trường hợp trên thì là true)*

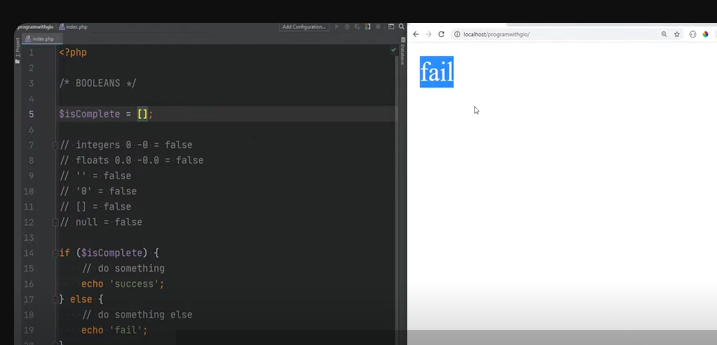


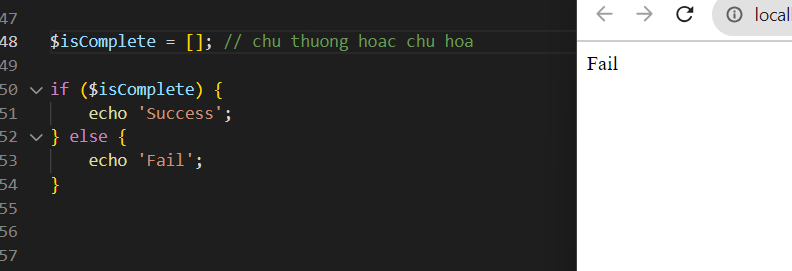


* We can set this to negative 0 and this will evaluate to false and it will print fail
* *Nếu đặt là âm 0 thì nó sẽ đánh giá là false và in ra fail*

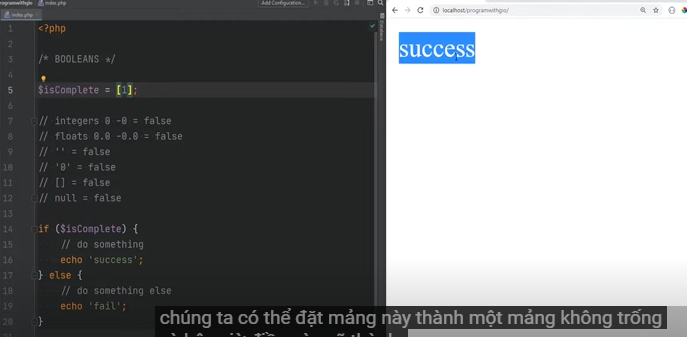


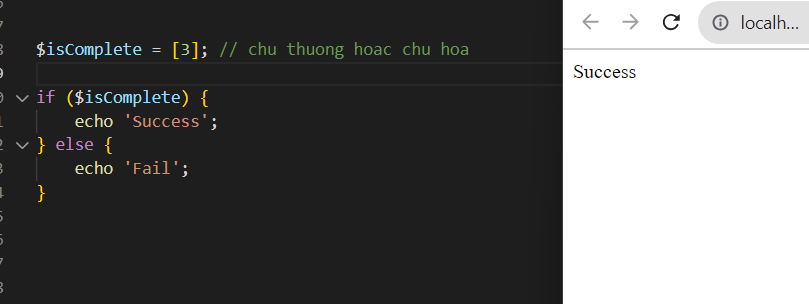
* We can set this to an empty array and this will evaluate to fail
* *Chúng ta đặt đây là 1 mảng rỗng sẽ đánh giá là fail*



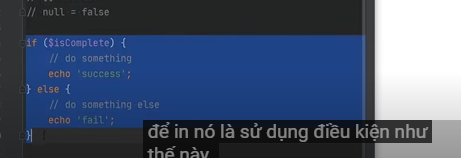


* As well we can set this to a non-empty array and this will be success
* *Đặt mảng này ko trống thì in ra success*

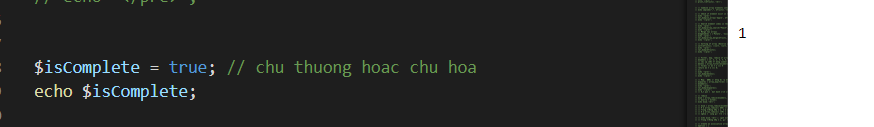
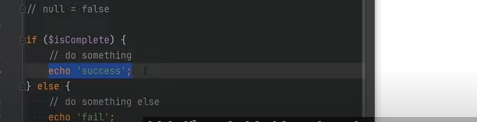




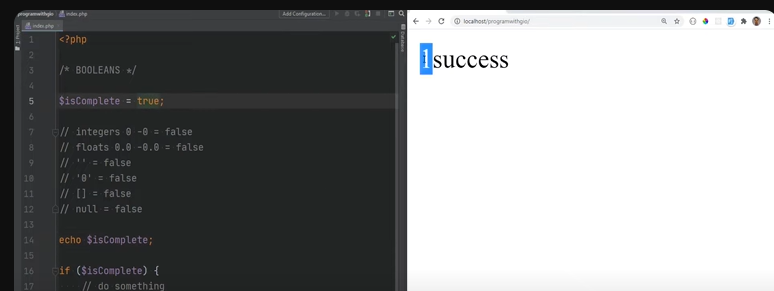
* Now how can you actually print the boolean now one way to print it is by using conditional like this
* *Làm thế nào để in ra boolean, 1 cách để in là dùng điều kiện như này*



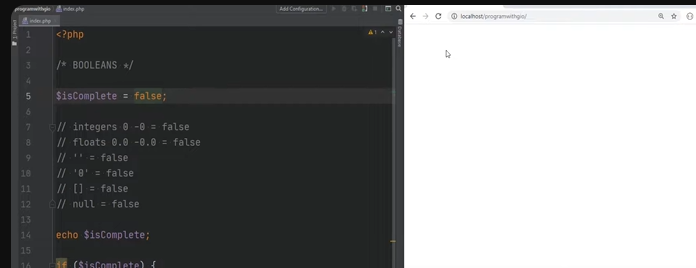
* where you don't actually print a boolean but instead when something is true you print something else like this for example
* *Khi bạn ko thực sự in boolean nhưng thay vào đó nếu một điều gì đó là true, bạn in ra một cái gì đó khác  như thế này*

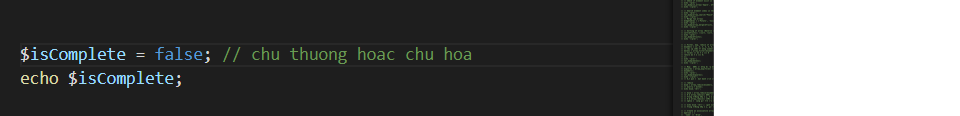


* Another way that you've seen before is just simply echo out the boolean so if we echo out is complete and set this to true it will echo out as one
* *Một cách khác mà bạn đã thấy trước đây là đơn giản là lặp lại boolean vì vậy nếu chúng ta lặp lại là complete và đặt isComplete là true và nó sẽ in ra 1*



* Let me comment this out for now
* *Comment lệnh ko cần thiết vào để in ra cho đỡ rối*
* And we see one if we set this to false we see nothing because it's just an empty string
* *Nếu đặt isComplete là false sẽ ko in vì nó là 1 chuỗi trống*



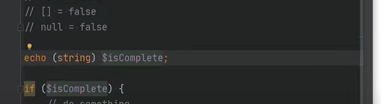


🡪Không in ra gì khi đặt false

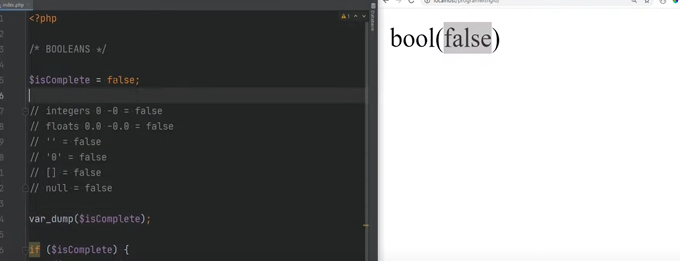
* Now you saw this in previous video
* *Bạn đã thấy điều này trong video trước*
* But why does it actually print an empty string for false and why does it print

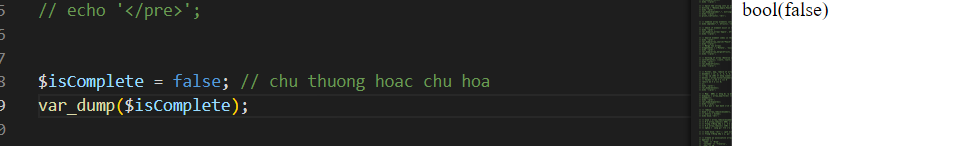
one for true

* *Nhưng tại sao nó thực sự in 1 chuỗi trống cho sai và tại sao nó in 1 chuỗi cho đúng*
* The reason for that is because when you're printing something on php it will try to cast it into a string so it's essentially doing same thing as this right here it's just you don't need to specify string in here
* *Lí do, khi bạn in in 1 cái gì đó trên php nó sẽ cố gắng chuyển thành 1 chuỗi về cơ bản nó thực hiện điều tương tự ngay tại đây chỉ là bạn ko cần chỉ định chuỗi ở đây*

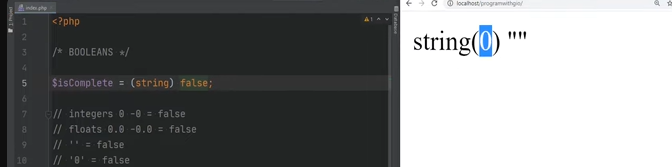


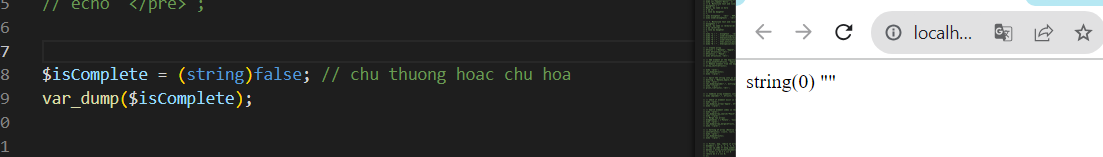
* And when boolean is cast into a string false becomes an empty string and true becomes one and we can check that by using var dom so let's do var dump is complete and we see that it's boolean type with the false
* *Và khi boolean được chuyển thành 1 chuỗi sai, trở thành 1 chuỗi rỗng và đúng trở thành 1 vì vậy kiểm tra bằng cách sử dụng var\_dump(isComplete) và thấy nó in là bool với false*



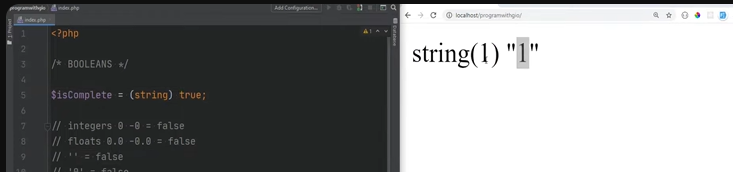


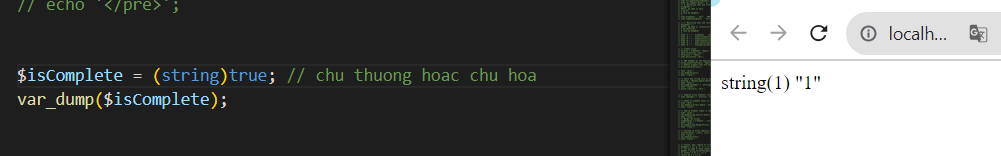
* And now let's convert it to string by using the cast and we see that it's a string type with zero characters in it so that's what happens when you do echo is complete
* *Và bây giờ chuyển đổi nó thành string bằng cách sử dụng lệnh cast và chúng ta sẽ thấy rằng đó là một chuỗi với không ký tự trong đó. Đó là điều xảy ra khi bạn sử dụng lệnh echo để in ra is\_complete.(False in ra 0)*





* And if we do this to true this becomes string with the value of 1
* *Và nếu chúng ta làm điều này với true, nó sẽ trở thành một chuỗi với giá trị là "1" (True in ra 1)*

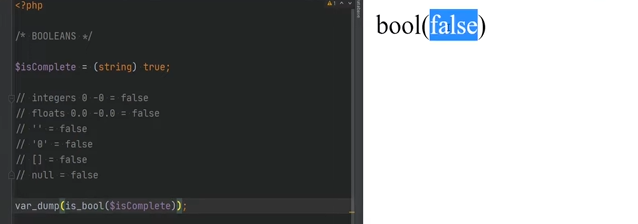


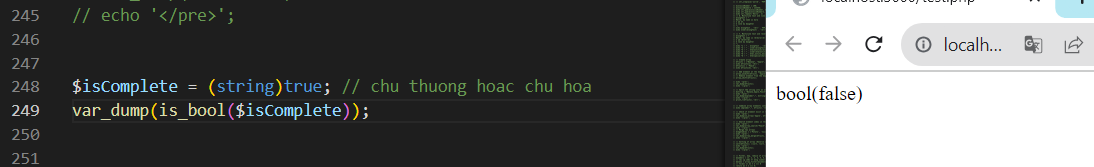


* So as you know var\_dump is one way to check the type whether it's string or integer or boolean
* *Như bạn đã biết, var\_dump là một cách để kiểm tra kiểu dữ liệu, xem liệu nó là chuỗi, số nguyên hay boolean*

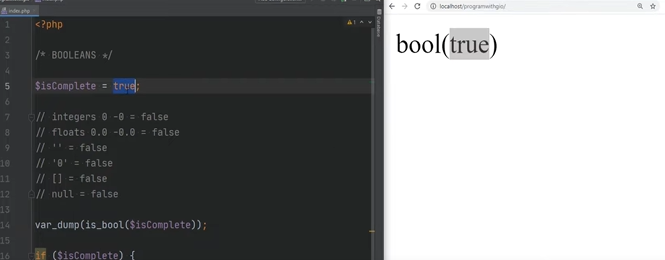


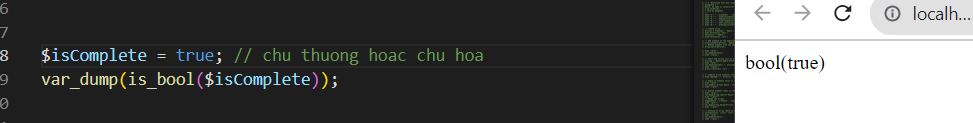
* Another way to check type for a boolean is by using a function called is\_bool and that will return true or false depending on whether the variable is true or false so in this case if we var\_dump the value of is\_bool it's going to return false because we're casting it into string
* *Một cách khác để kiểm tra kiểu dữ liệu cho một boolean là sử dụng hàm có tên là is\_bool nó sẽ trả về true hoặc false tùy thuộc vào biến có giá trị true hay false. Vì vậy, trong trường hợp này, nếu chúng ta var\_dump giá trị của isbul, nó sẽ trả về false vì chúng ta đã chuyển đổi nó thành chuỗi.*



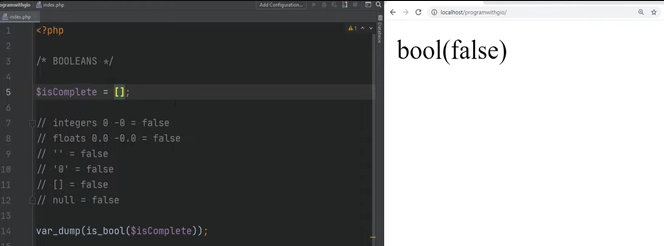


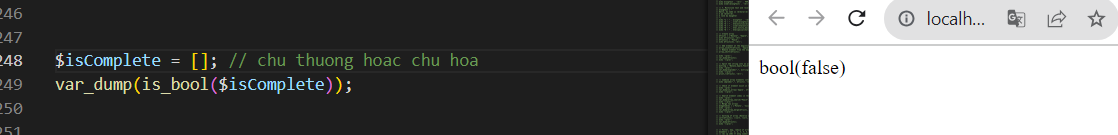
* If we change this back to true now it will return true because it's a boolean value
* *Nếu chúng ta đổi trở lại true, bây giờ nó sẽ trả về true vì đó là một giá trị boolean.*



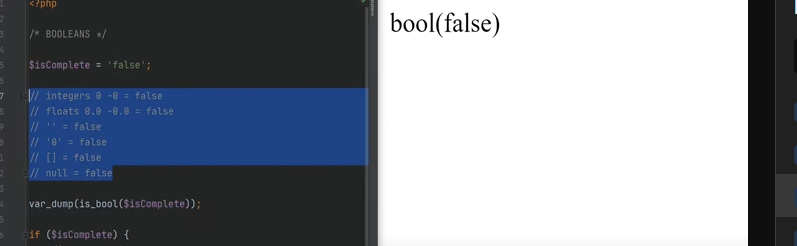


* If we change this to an empty array this is false
* *Nếu chúng ta đặt thành một mảng rỗng, nó sẽ trả về false.*

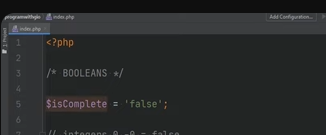




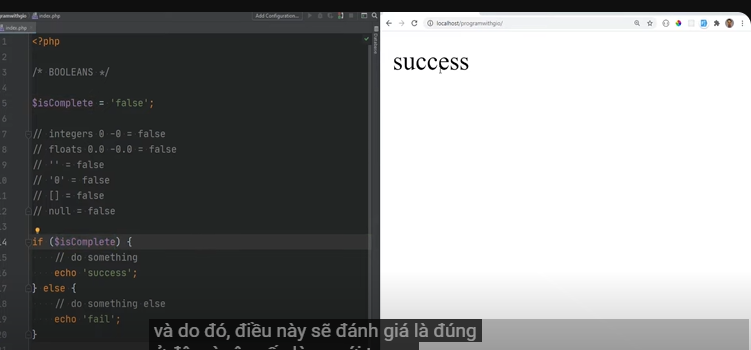
* And one more thing i want to mention before wrapping up this video is that if you have a string that's equal to false you might think that this would evaluate to false but it would actually evaluate to true because as mentioned right in here anything that's not an empty string or a zero it will evaluate to true
* *Một điều quan trọng khác mà tôi muốn đề cập trước khi kết thúc video này là nếu bạn có một chuỗi có giá trị bằng false, bạn có thể nghĩ rằng điều này sẽ đánh giá là false nhưng thực ra nó sẽ đánh giá là true Bởi vì như đã đề cập ở trên, bất cứ điều gì không phải là một chuỗi rỗng hoặc số không, nó sẽ được đánh giá là true*

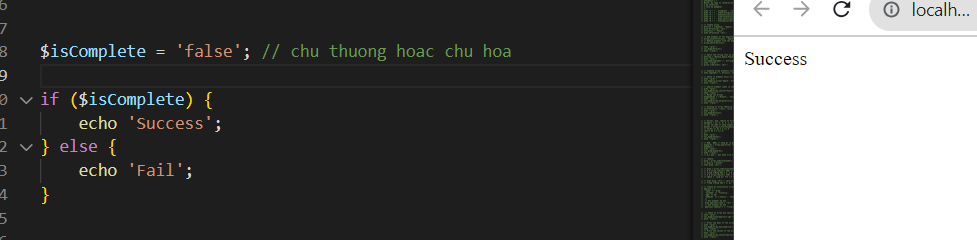


* So even though this says false here this is not an empty string and it's not a zero and therefore this will evaluate to true
* *Vì vậy, mặc dù ở đây nó nói là false, nhưng điều này không phải là một chuỗi rỗng và nó cũng không phải là số không, do đó nó sẽ đánh giá là true.*

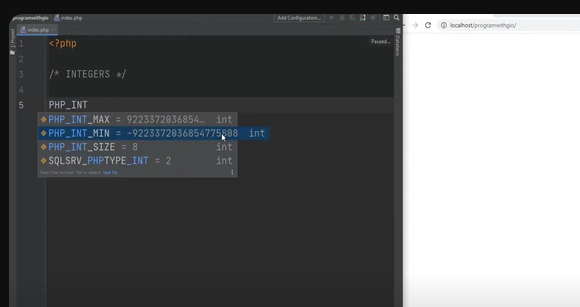


* In here so if we refresh the page we see success
* *Do đó, nếu chúng ta làm mới trang, chúng ta sẽ thấy "success*

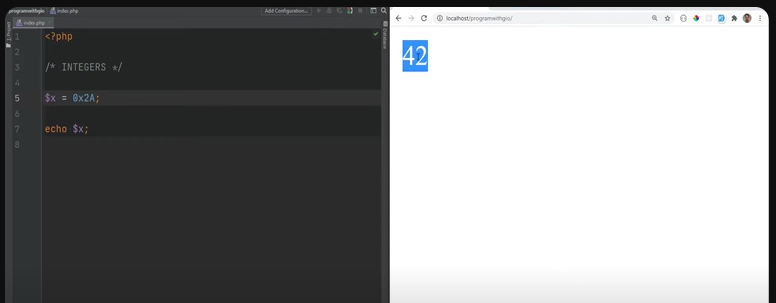


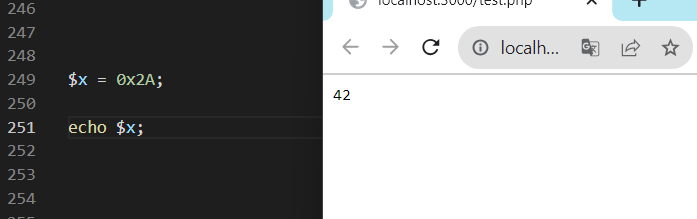


* 1. **PHP Interger Data Type – Full PHP 8 Tutorial**
* Integers are numbers without any decimal points like 1 2 3 0 negative 5 and so on the size of an integer though depends on the platform for 32-bit systems it's from negative 2 billion to positive 2 billion and for 64 bits it's much more and you can check that easily by using the predefined constants that we covered in the previous videos
* *Số nguyên là các số không có phần thập phân, chẳng hạn như 1, 2, 3, 0, -5 và cứ tiếp tục như vậy. Kích thước của một số nguyên phụ thuộc vào nền tảng. Trên hệ thống 32-bit, nó nằm trong khoảng từ -2 tỷ đến +2 tỷ. Với 64-bit, giới hạn sẽ lớn hơn nhiều và bạn có thể kiểm tra dễ dàng bằng cách sử dụng các hằng số được định nghĩa sẵn mà chúng ta đã bàn luận trong các video trước đó*
* And those are php underscore and max min and size and for my platform these are maximum and minimum values
* *Và đó là các hằng số PHP gạch dưới tối thiểu và kích thước tối đa và đối với nền tảng của tôi đây là các số nguyên có giá trị tối đa và tối thiểu*

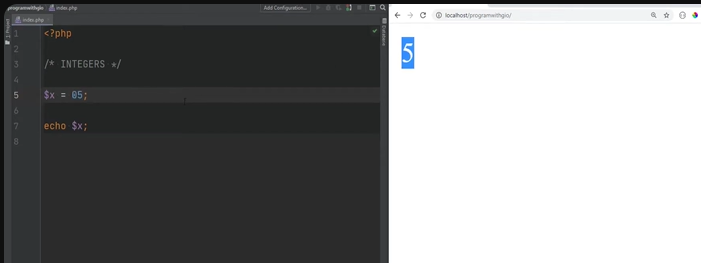


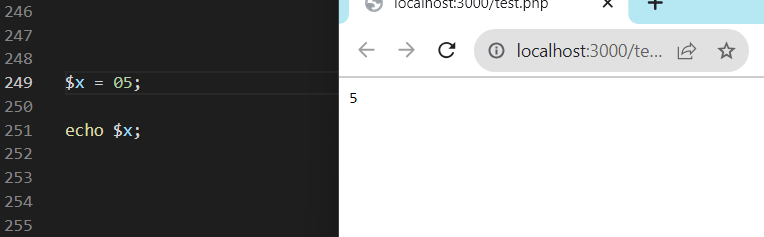
* Integers can be specified in decimal or base 10 which is for example like this this is base 10 or this is base 10 and then you could echo it out and it will just display five it can also be defined as a hexadecimal which is base 16 and hexadecimal numbers are prefixed by zero x so it would be zero x to a and you could print that out which is the decimal value is 42.
* *Số nguyên có thể được chỉ định dưới dạng thập phân hoặc cơ số 10 ví dụ như thế này đây là cơ số 10 bạn có thể in ra giá trị đó bằng lệnh echo và nó sẽ hiển thị số 5 nó cũng có thể được định nghĩa dưới dạng số thập lục phân hoặc cơ số 16 Số thập lục phân có tiền tố là "0x" vì vậy chẳng hạn "0x2A" sẽ được in ra và giá trị thập phân tương ứng là 42*

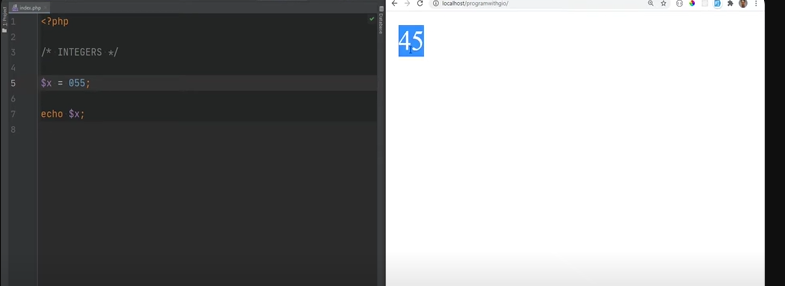


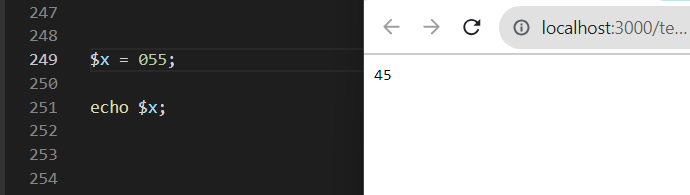


* You could also define integers as octal numbers and octal numbers are prefixed by zero so zero five would evaluate to five zero fifty five for example would evaluate to 45.
* *Bạn cũng có thể định nghĩa các Số nguyên dưới dạng số bát phân số bát phân có tiền tố là "0" vì vậy chẳng hạn "05" sẽ được chuyển đổi thành số 5 "055" sẽ được chuyển đổi thành số 45.*

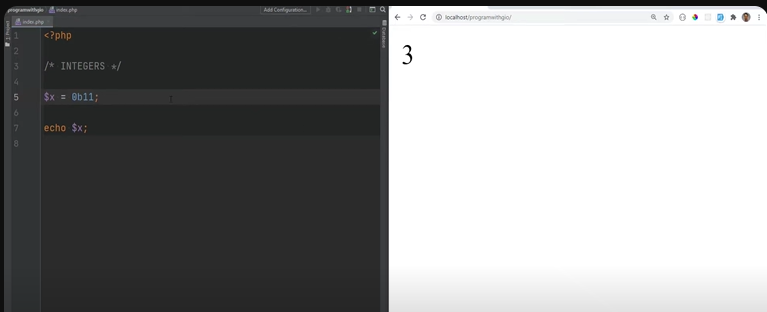


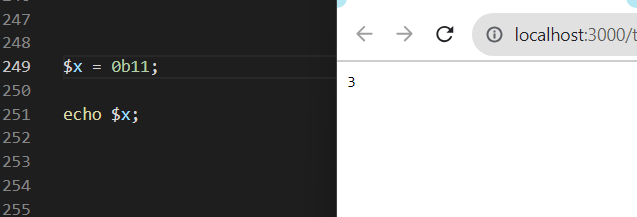


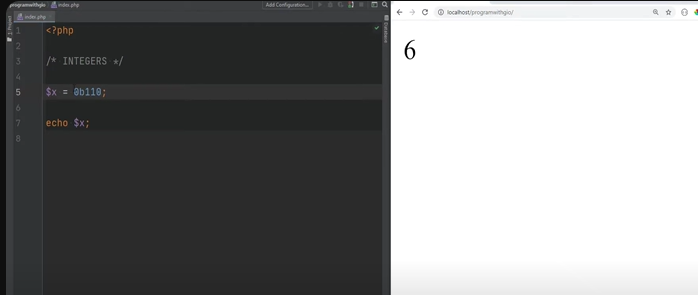


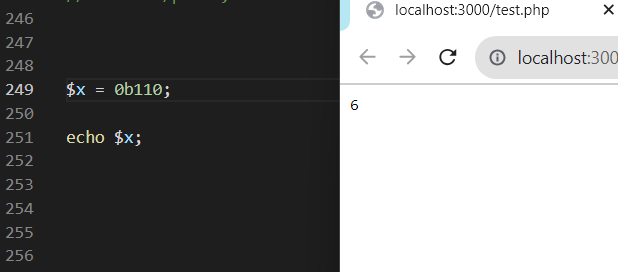


* It could also represent integers as binary numbers which are prefixed by zero b so zero b one one would equal to three and one one zero would equal to 6.
* *Bạn cũng có thể biểu diễn số nguyên dưới dạng số nhị phân, có tiền tố là "0b"  Ví dụ, "0b11" sẽ tương đương với số 3 và "0b110" tương đương với số 6*

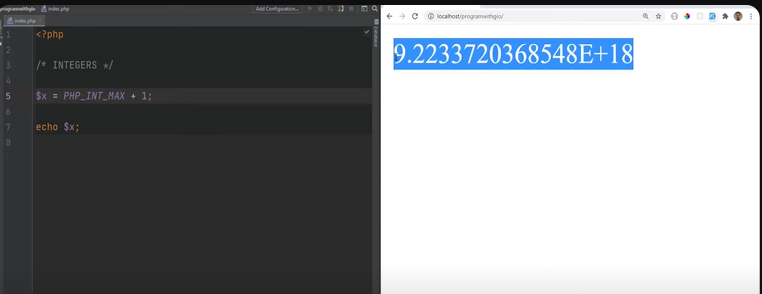


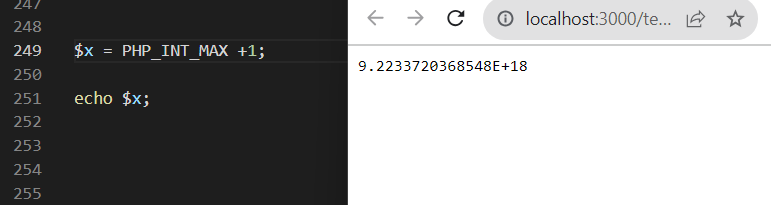




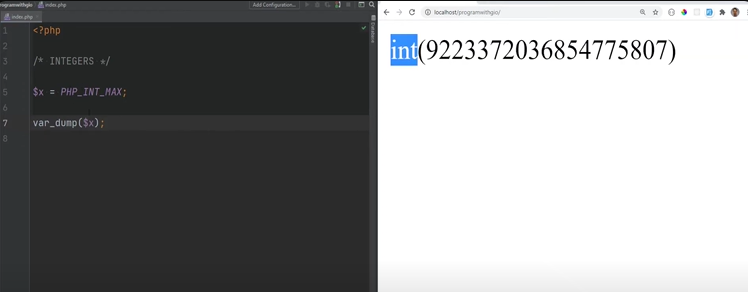


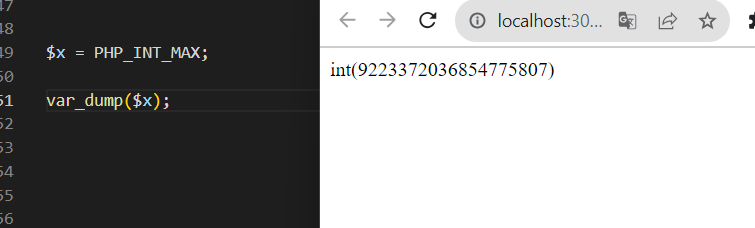
* Now what happens when an integer overflow occurs
* Nếu một số nguyên vượt quá giới hạn, điều gì sẽ xảy ra?
* So for example we have php and max this value and we echo it out and this is the value that we have but if we were to add 1 to this let's see what happens the value changes but the type of it is no longer integer
* *VD, nếu chúng ta có giá trị tối đa của PHP và in ra nó, giá trị đó sẽ hiển thị Nhưng nếu chúng ta thêm 1 vào giá trị này, xem sẽ xảy ra gì. Giá trị thay đổi, nhưng kiểu dữ liệu không còn là số nguyên nữa.*



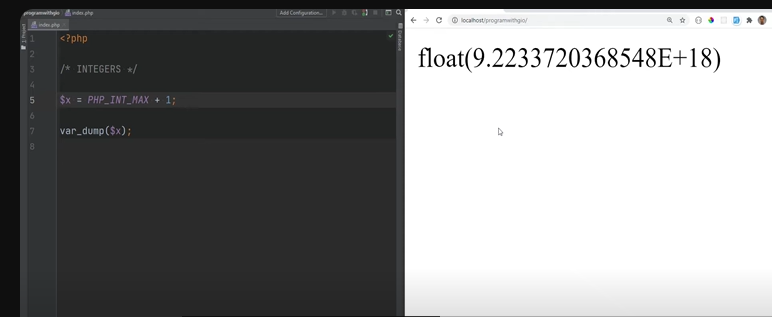


* And we can check that easily by using var\_dump so let's do var dump x and that is the type integer
* *Chúng ta có thể kiểm tra điều đó dễ dàng bằng cách sử dụng var\_dump. Vì vậy, hãy thử var\_dump(x), nơi x là giá trị trước khi thêm 1 vào. Ban đầu, kiểu dữ liệu là số nguyên,*

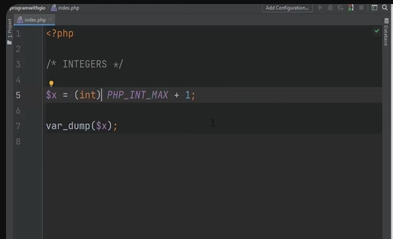


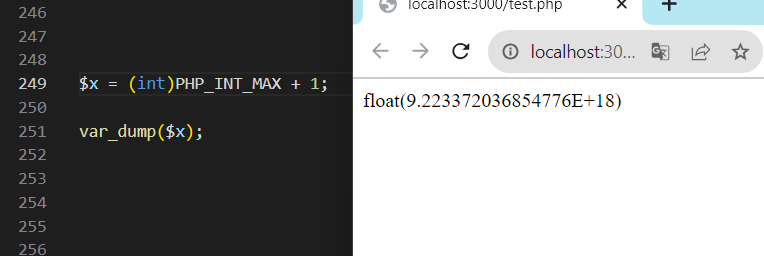


* Now if we're going to add 1 to this the type changes the float .So when integer goes out of bounds it will Get changed into the float data type and we'll talk about the flow data type in the next video
* *Nhưng sau khi thêm 1, kiểu dữ liệu trở thành số thực (float). Nên sau khi số nguyên vượt quá giới hạn nó sẽ được đổi thành kiểu dữ liệu float và Chúng ta sẽ nói về kiểu dữ liệu số thực trong video tiếp theo.*

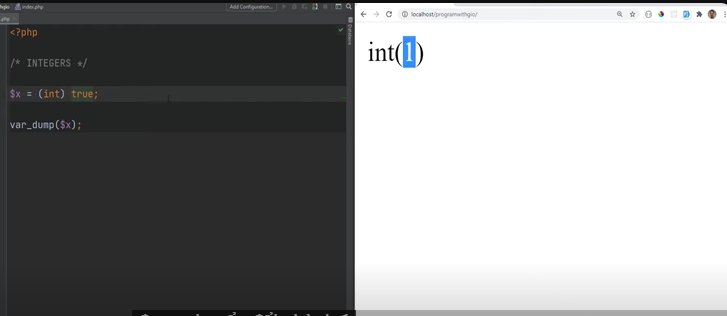


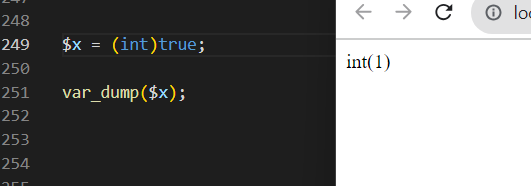
* Now let's talk about casting you could cast values using parenthesis int or parentheses integer and this would both cast the value into an integer
* *Giờ hãy nói về việc ép kiểu (casting). Bạn có thể ép kiểu giá trị bằng cách sử dụng dấu ngoặc đơn và từ khóa int hoặc dấu ngoặc đơn và từ khóa integer. Cả hai cách này đều chuyển đổi giá trị thành số nguyên*

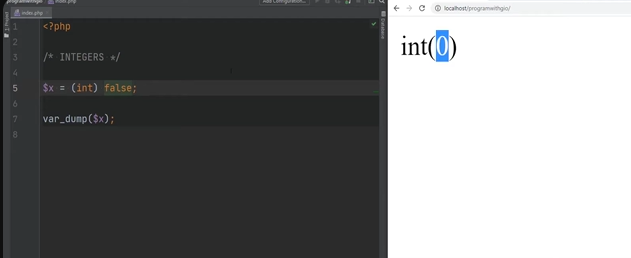


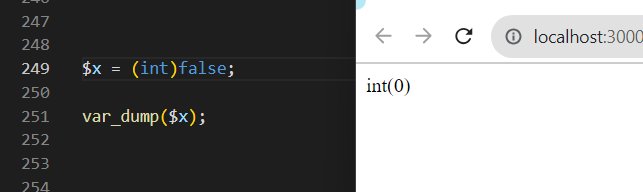


* Let's review how the other data types are cast into integers boolean true when cast to integer will get converted to 1 .boolean false when caster integer will get converted to zero
* *Hãy xem cách ép kiểu các kiểu dữ liệu khác thành số nguyên Khi ép kiểu boolean true thành số nguyên, nó sẽ chuyển đổi thành 1*. Khi ép kiểu boolean false thành số nguyên, nó sẽ chuyển đổi thành 0.

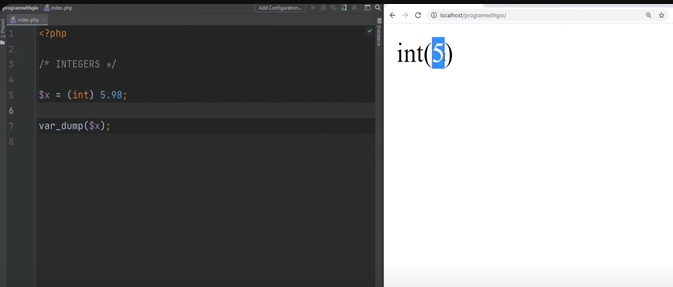


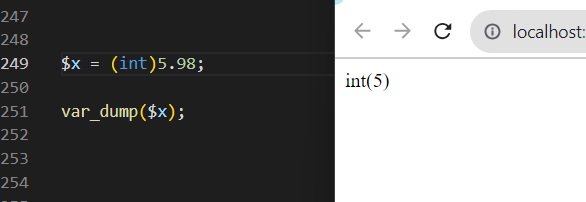




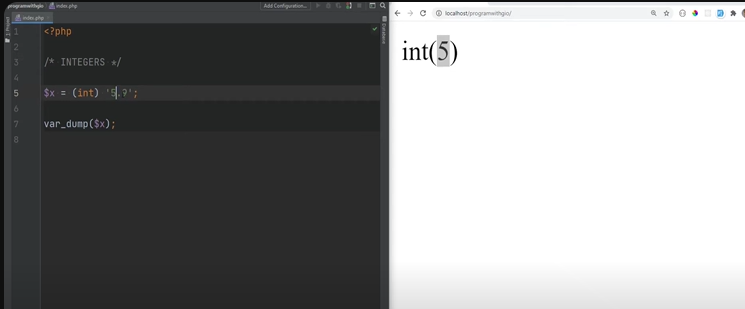


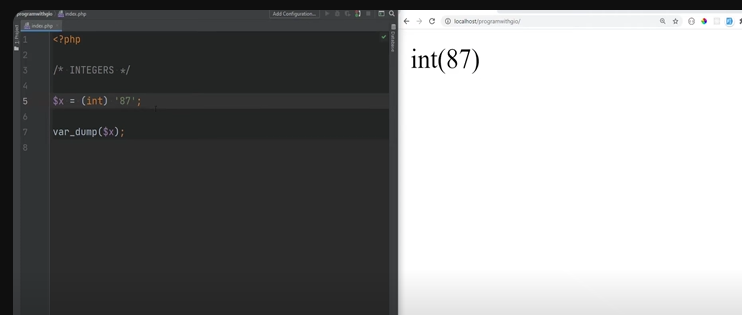
* The floating number will lose the decimal points so this will basically get rounded down to 5 so even though it's 0.98 it's not going to be rounded up it will be rounded down so essentially you're just losing this value here so this will result in 5.
* *Số thực sẽ mất phần thập phân, vì vậy nó sẽ được làm tròn xuống. Ví dụ, số 5.9 sẽ được chuyển đổi thành số 5. Mặc dù nó là 0.98, nhưng nó sẽ không được làm tròn lên, mà sẽ bị làm tròn xuống thành số 5.*

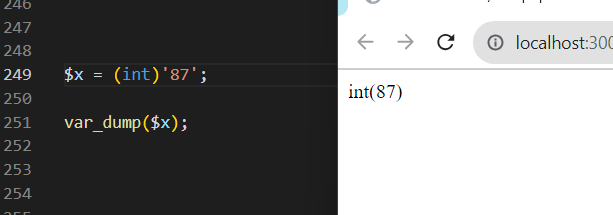




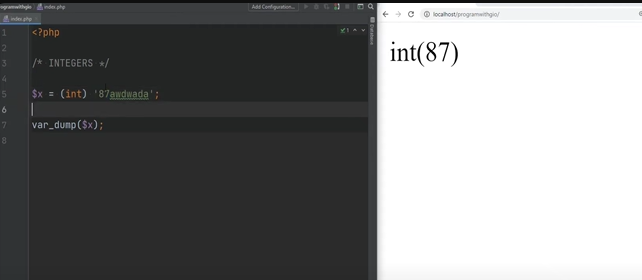
* When strings are being converted to integers if the string is numeric something like 5.9 this will get cast into 5 as well because the 0.9 will get removed
* *Khi chuỗi được chuyển đổi thành số nguyên, nếu chuỗi là một số, ví dụ như "5.9", nó sẽ được chuyển đổi thành số nguyên 5, vì phần thập phân sẽ bị loại bỏ.(Chuỗi cũng giống số nguyên bên trên)*

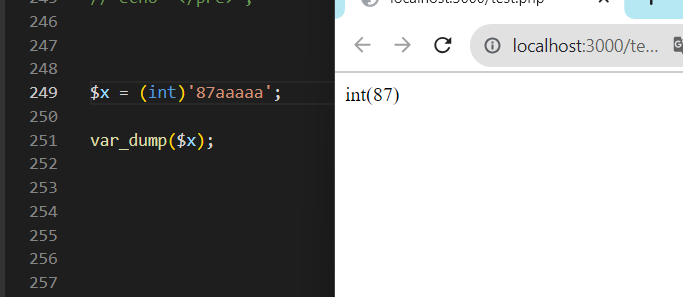


* If the string is just another numeric integer for example it will get cast into an integer normally
* *Nếu chuỗi chỉ chứa một số nguyên, chẳng hạn "87", nó sẽ được chuyển đổi thành số nguyên bình thường*

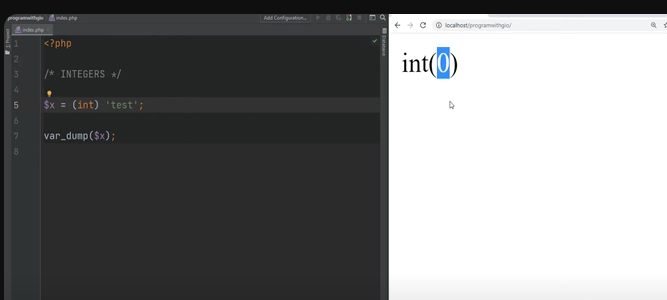


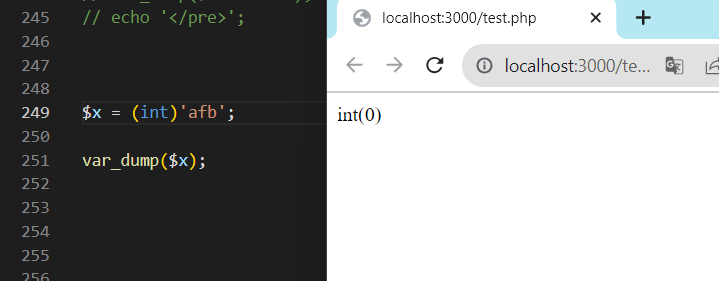
* In số nguyên bình thường
* If the string contains some kind of text with the number when it's cast to integer php will attempt to convert this and resolve it as an integer which means that it will just return an 87 out of this
* *Nếu chuỗi chứa văn bản kèm theo số, khi chuyển đổi thành số nguyên, PHP sẽ cố gắng chuyển đổi và giải quyết nó thành số nguyên. Điều này có nghĩa là nó sẽ trả về 87 trong trường hợp này*

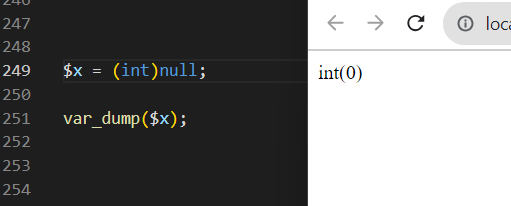




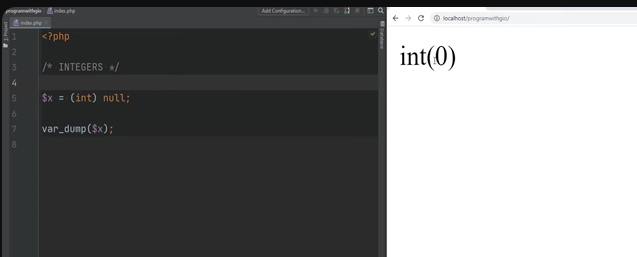
* However if the string cannot be resolved as integer it will just be converted to 0 so for example if you were to convert this to an integer it will result in zero
* *Tuy nhiên, nếu chuỗi không thể giải quyết thành số nguyên, nó sẽ được chuyển đổi thành 0. Ví dụ, nếu bạn chuyển đổi chuỗi "test" thành số nguyên, nó sẽ trở thành 0.*



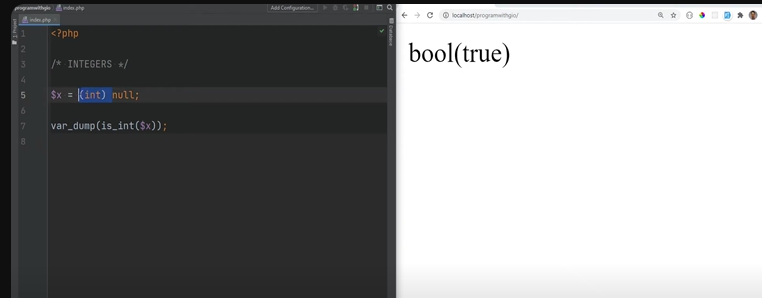


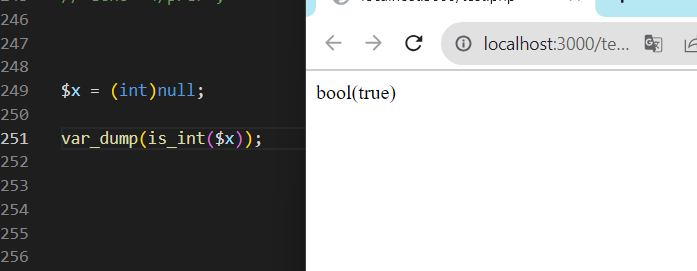


* And when converting nulls to integers it will also be converted to 0.
* *Khi chuyển đổi giá trị null thành số nguyên, nó cũng sẽ được chuyển đổi thành 0*

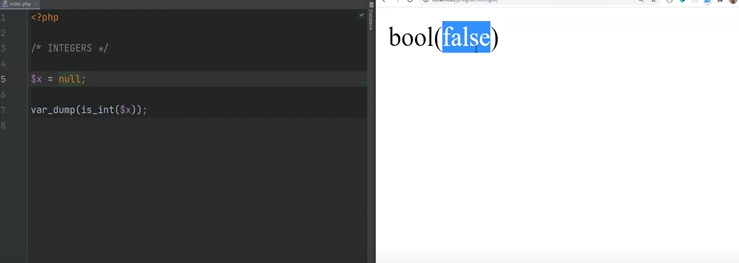


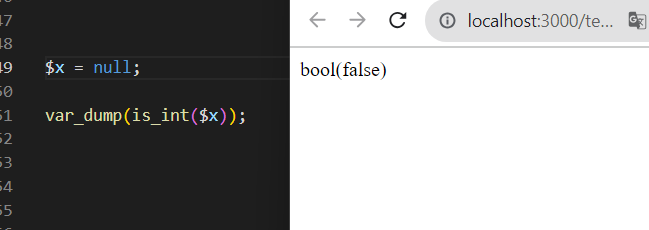
* Now the way you could check if the Variable is an integer is by using function called is int or is integer and that will return true or false if the variable is integer or not so if we give it x here this will return true because we're casting it to an integer
* *Bạn có thể kiểm tra xem một biến có phải là số nguyên hay không bằng cách sử dụng hàm is\_int hoặc is\_integer. Hàm này sẽ trả về true hoặc false tùy thuộc vào biến có phải là số nguyên hay không. Ví dụ, nếu bạn đưa vào biến x, hàm is\_int(x) sẽ trả về true vì chúng ta đang ép kiểu nó thành số nguyên*



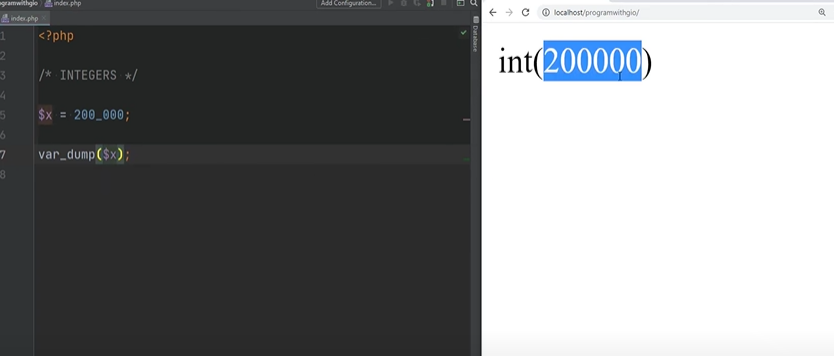


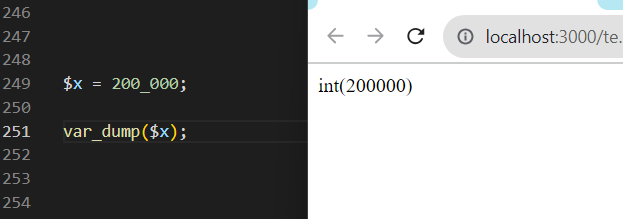
* But if we remove this and we run it again it returns false because it's not an integer
* *Nhưng nếu bạn loại bỏ phần ép kiểu và chạy lại, nó sẽ trả về false vì nó không phải là số nguyên*



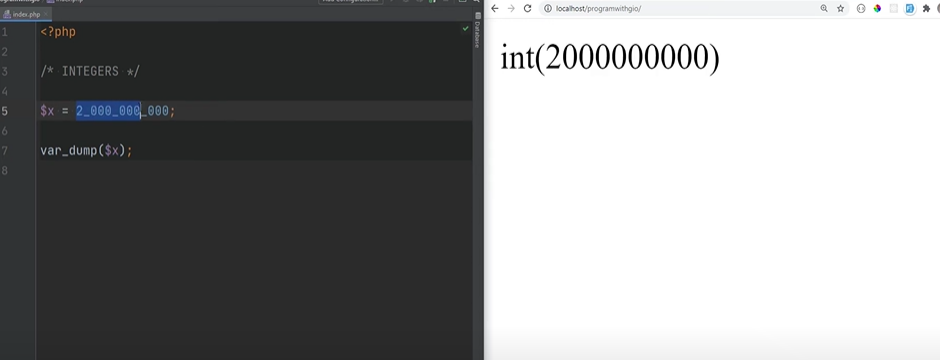


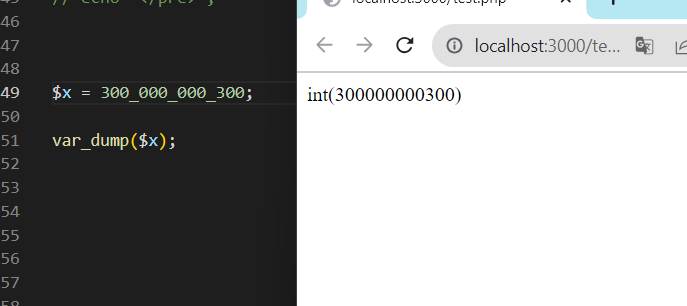
* Also as of php 7.4 for readability purposes you could use underscores within your integer so for example you could have something like 200 000 and now you could actually do something like this which makes it more readable so if we refresh and let's remove the async check here refresh and we get to 100 000
* *Kể từ phiên bản PHP 7.4, bạn có thể sử dụng dấu gạch dưới để làm cho số nguyên dễ đọc hơn. Ví dụ, bạn có thể có một số như 200.000 và giờ bạn có thể viết như sau để làm cho nó dễ đọc hơn: 200\_000. Khi làm vậy, nếu làm mới trang và loại bỏ kiểm tra bất đồng bộ ở đây, sau đó làm mới, bạn sẽ thấy giá trị là 100.000*

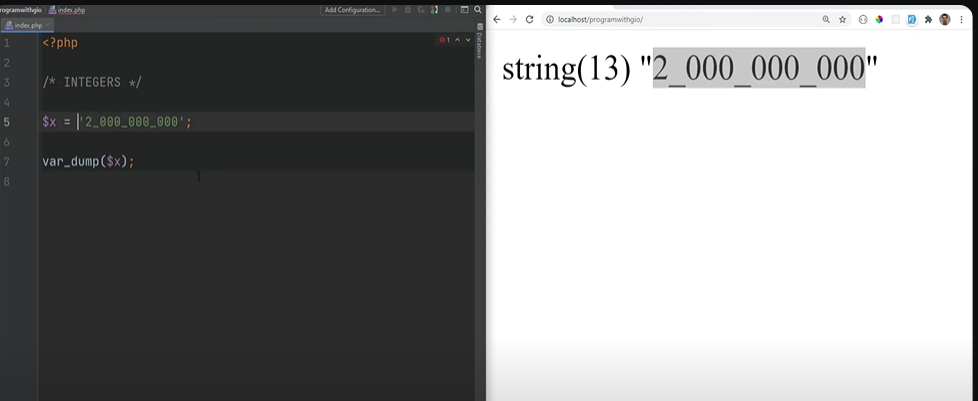


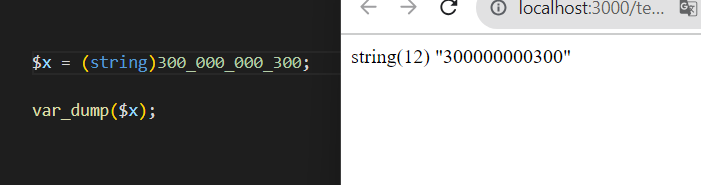


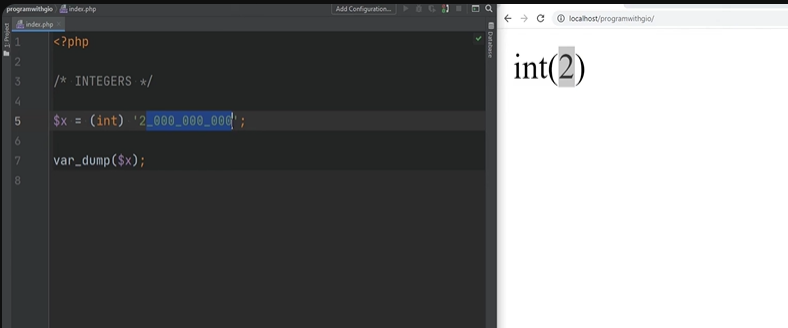
* And this could be useful for long numbers something like this and it just adds a readability to the number without using commas or anything like that the underscores are basically just uh removed and ignored by the php however if you turn this into a string this will not work this will basically just treat this as a string and not as an integer
* *Điều này cũng có thể hữu ích cho các số lớn. và nó chỉ tăng thêm khả năng đọc cho số mà ko cần bằng cách sử dụng dấu phẩy hoặc bất cứ thứ gì ,tương tự Dấu gạch dưới được loại bỏ và bị bỏ qua bởi PHP Tuy nhiên, nếu bạn chuyển đổi nó thành một chuỗi, điều này sẽ không hoạt động và nó sẽ chỉ coi đó là một chuỗi và không phải là một số nguyên*

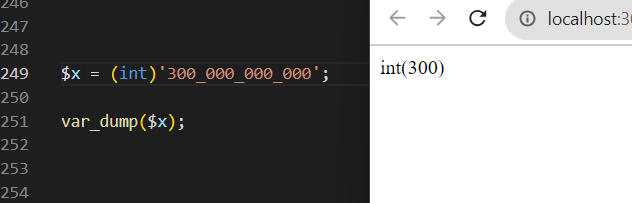








* And then if you cast this to an end this will just be casted as two because anything after that will just get removed
* *Và sau đó, nếu bạn ép kiểu này thành một số nguyên, nó sẽ chỉ được ép kiểu thành hai vì bất kỳ điều gì sau đó sẽ bị loại bỏ* 



* So keep that in mind if you do use underscores it just has to be integer representation and not a string
* *Vì vậy, hãy nhớ rằng nếu bạn sử dụng dấu gạch dưới, nó chỉ có thể là biểu diễn số nguyên và không phải là một chuỗi*