**1.6:**

Số nguyên là các số không có bất kỳ dấu thập phân nào như 1; 2; 3; -5,…… tùy thuộc vào kích thước của số nguyên.

Đối với hệ thống 32 bit, nó từ âm 2 tỷ đến dương 2 tỷ. Đối với 64 bit thì nhiều hơn thế nữa nên bạn có thể kiểm tra điều đó một cách dễ dàng bằng cách sử dụng các hằng số được xác định trước mà tôi đã đề cập trong các video trước và đó là các hằng số PHP\_INT\_MAX hoặc MIN hoặc SIZE.

Đây là các số nguyên có giá trị tối đa và tối thiểu có thể được chỉ định ở dạng thập phân hoặc cơ số 10, nghĩa là ví dụ như thế này đây là cơ số 10

A screenshot of a computer

Description automatically generated

hoặc đây là cơ số 10 và sau đó bạn có thể lặp lại và nó sẽ chỉ hiển thị 5

A black and white background

Description automatically generated with medium confidence Nó cũng có thể được định nghĩa là hệ thập lục phân cơ số 16 và các số thập lục phân có tiền tố là 0x nên nó sẽ là 0x2A và bạn có thể in ra giá trị thập phân là 42.

A black and white line

Description automatically generated with medium confidence

Bạn cũng có thể định nghĩa các số nguyên là số bát phân và số bát phân có tiền tố là 0 nên số 05 sẽ có giá trị là 05; 055 sẽ bằng 45.

A black and white line

Description automatically generated

nó cũng có thể biểu diễn các số nguyên dưới dạng số nhị phân có tiền tố là 0b nên 0b11 sẽ bằng 3

A black and white image

Description automatically generated with medium confidence

và 0b110 sẽ bằng 6.

A black and white line

Description automatically generated with medium confidencea

Bây giờ điều gì xảy ra khi xảy ra tràn số nguyên, ví dụ như chúng ta có PHP\_INT\_MAX này và chúng tôi lặp lại nó và đây là giá trị mà chúng tôi có

A screenshot of a computer

Description automatically generated

Nhưng nếu chúng tôi thêm 1 vào đây, hãy xem điều gì xảy ra

A screenshot of a computer

Description automatically generated

thì giá trị sẽ thay đổi nhưng loại của nó không còn là số nguyên

và chúng tôi có thể kiểm tra điều đó một cách dễ dàng bằng cách sử dụng

var\_dump($x), vì vậy hãy lấy var\_dump($x) và bây giờ đó là kiểu số nguyên

A close up of a computer screen

Description automatically generated

nếu chúng ta thêm 1 vào kiểu này thì kiểu này sẽ thay đổi số float nên

khi số nguyên vượt quá giới hạn, nó sẽ được đổi thành kiểu dữ liệu float

A close up of a computer screen

Description automatically generated

Và chúng ta sẽ nói về dữ liệu luồng ở video tiếp theo.

Bây giờ hãy nói về việc truyền bạn có thể truyền các giá trị bằng cách sử dụng

(int) hoặc (integer) và cả hai sẽ chuyển giá trị thành một số nguyên. Hãy

xem lại cách các loại dữ liệu khác được chuyển thành số nguyên

bool true khi chuyển thành số nguyên sẽ được chuyển đổi thành 1

A black and white line

Description automatically generated

bool false khi chuyển thành số nguyên sẽ được chuyển đổi thành 0,

A black and white line

Description automatically generated

số float sẽ mất điểm thập phân( decimal points) nên về cơ bản, số này sẽ được làm tròn xuống 5 nên mặc dù là 0,98 nhưng nó sẽ không được làm tròn lên nhưng về cơ bản nó sẽ được làm tròn xuống vì vậy về cơ bản,bạn chỉ mất giá trị này ở đây nên

kết quả là 5.

A black rectangular object with a white stripe

Description automatically generated

khi chuỗi được chuyển đổi thành số nguyên nếu chuỗi đó là số giống như 5,9 thì chuỗi này cũng sẽ được chuyển thành 5 vì 0,9 sẽ bị xóa nếu chuỗi chỉ là một số nguyên khác

A black and white line

Description automatically generated

ví dụ: thông thường nó sẽ được chuyển thành một số nguyên

nếu chuỗi chứa một số loại văn bản có số khi nó được chuyển thành số nguyên.

php sẽ cố gắng chuyển đổi số này và phân giải nó thành một số nguyên, điều đó có nghĩa là nó sẽ chỉ trả về 87 trong số này

A black and white line

Description automatically generated with medium confidence

Tuy nhiên nếu chuỗi không thể được phân giải dưới dạng số nguyên thì nó

sẽ chỉ được chuyển đổi thành 0, vì vậy,

ví dụ:

Nếu bạn chuyển đổi chuỗi này thành int\_test thì nó sẽ có kết quả bằng 0

A black and white image

Description automatically generated with medium confidence

Và khi chuyển đổi null thành số nguyên, nó cũng sẽ được chuyển đổi thành 0.

A black and white line

Description automatically generated with medium confidence

bây giờ là cách bạn có thể kiểm tra xem biến đó có phải là số nguyên hay không bằng cách sử dụng

hàm có tên là is\_int hay là integer và nó sẽ trả về đúng hoặc sai nếu biến đó là số nguyên hay không,

Vì vậy nếu chúng ta cung cấp cho nó x ở đây thì điều này sẽ trả về bool(true)

A black rectangular object with a white border

Description automatically generated  
 Vì chúng ta đang truyền nó thành một số nguyên nhưng nếu chúng tôi loại bỏ số này và chạy lại thì nó sẽ trả về bool(false)

A black and white striped background

Description automatically generated with medium confidence

Kể từ php 7.4 vì mục đích dễ đọc, bạn có thể sử dụng dấu gạch dưới trong số nguyên của mình, ví dụ như bạn có thể có số nguyên như 200 000 và bây giờ bạn có thể thực sự làm một cái gì đó như thế này để làm cho nó dễ đọc hơn

A computer screen shot of a black background

Description automatically generated

Vì vậy nếu chúng ta làm mới và hãy xóa kiểm tra không đồng bộ ở đây, hãy

làm mới và chúng ta đạt tới 200 000.

A black rectangular object with a white stripe

Description automatically generated

và điều này có thể hữu ích cho các số dài như thế này

A black screen with numbers and text

Description automatically generated

và nó chỉ tăng thêm khả năng đọc cho số mà không cần bằng cách sử dụng dấu phẩy hoặc bất cứ thứ gì tương tự, dấu gạch dưới về cơ bản chỉ bị php loại bỏ và bỏ qua, tuy nhiên nếu bạn biến chuỗi này thành sring thì nó sẽ không hoạt động thì về cơ bản nó sẽ chỉ coi đây là một chuỗi chứ không phải là một số nguyên và sau đó nếu bạn sử dụng chuỗi này

A screen shot of a computer

Description automatically generated

Cuối cùng, nó sẽ chỉ được phân thành 2 vì mọi thứ sau đó sẽ bị xóa, vì vậy hãy ghi nhớ điều đó nếu bạn sử dụng dấu gạch dưới thì nó chỉ là biểu diễn số nguyên chứ không phải một chuỗi. Đây là cho video tiếp theo Chúng ta sẽ nói về float và có một số điểm rất quan trọng về cách làm việc với float cũng như một số điều cần chú ý, vì vậy hãy chú ý theo dõi và nhớ xem video đó, vui lòng thích video này chia sẻ và đăng ký và tôi sẽ gặp bạn ở lần tiếp theo

**1.7:**

Số dấu phẩy động là những số có dấu thập phân nên chúng ta có thể có x=13,5 và nếu lặp lại x chúng ta nhận được 13,5

Nó cũng có thể biểu thị các số thực ở dạng mũ nên:

Ví dụ chúng ta có thể biến số này thành 13,5e3 và bằng

13 500. Chúng ta cũng có thể tính số âm nên có thể tính 13.5e-3 và kết quả là 0,0135

Và kiểu của x là float, bây giờ chúng ta có thể kiểm tra điều đó bằng cách thực hiện var\_dump và xác nhận rằng đó là số thực ngay cả khi chúng tôi làm với số dương và nó không thực sự chứa số thập phân, nó vẫn chảy như chúng ta đã thảo luận với các số nguyên kể từ php7.4

bạn thực sự có thể có dấu gạch dưới trong các số của mình để dễ đọc hơn nên bạn có thể có số như 13 000 và ở đây bạn có thể thêm dấu gạch dưới

A screenshot of a computer

Description automatically generated

và điều này hoàn toàn ổn và nếu chúng tôi làm mới thì bây giờ chúng tôi sẽ nổi, chúng tôi có thể xóa số mũ từ đây và làm mới và nó vẫn hoạt động

A screenshot of a computer

Description automatically generated

giống như số nguyên, kích thước của số thực phụ thuộc vào nền tảng

và bạn có thể kiểm tra điều đó bằng cách sử dụng hằng số được xác định trước

và các hằng số đó là php float max và php float min vì vậy hãy kiểm tra xem php

float max và min để bạn có thể sử dụng những hằng số này, hãy lặp lại mức tối đa

và chúng tôi thấy rằng đó là một con số khá lớn.

Nhận thức được khi làm việc với các số thực là

độ chính xác hạn chế của chúng, chẳng hạn nếu chúng ta lấy biểu thức này

ngay tại đây floor(0.1 +0.7)\*10 bạn nghĩ rằng kết quả này sẽ

trả về đúng 8 vì 0,1+ 0,7 tức là 0,8\* 10 là 8

và sau đó floor chỉ nên làm tròn xuống thành 8 và chỉ trả về 8

nếu làm mới, chúng ta nhận được 7 và lý do là vì 0,1

hoặc 0,7 không có biểu diễn chính xác vì số dấu phẩy động ở cơ số 2 là nhị phân

và nhị phân được sử dụng nội bộ để lưu trữ các số động và do đó khi chuyển đổi số này thành số nhị phân bên trong, nó sẽ mất một số độ chính xác

và trong trường hợp này (0,1+0,7) \*10

thực sự bằng với số này

A screenshot of a computer

Description automatically generated

Ngay tại đây và như bạn biết về cơ bản floor chỉ làm tròn tất cả các số xuống và trong trường hợp này khi áp dụng giá trị floor cho số này, nó chỉ loại bỏ toàn bộ số thập phân ở đây và chúng ta chỉ còn lại 7 và đó là lý do tại sao nó hiện 7 ở đây.

Bây giờ tôi biết rằng tôi đã giới thiệu rất nhiều khái niệm ở đây

chẳng hạn như phép cộng nhân, phương thức floor

, v.v. nhưng đừng lo lắng quá nhiều về điều này ngay bây giờ chúng ta sẽ

xem xét các toán tử trong một video riêng biệt.

Tôi chỉ muốn cho bạn biết những điều cần lưu ý khi làm việc với float để bạn biết floor chỉ làm tròn mọi thứ xuống và cái đối lập với cái đó được gọi là ceil hoặc seal, về cơ bản làm tròn mọi thứ và nếu chúng ta làm mới trang này, chúng ta sẽ nhận được 8 như mong đợi,

A black and white photo

Description automatically generated with medium confidence

bây giờ hãy làm một ví dụ khác với ceiling trong đó nó sẽ dẫn đến

giá trị không mong muốn vì vậy hãy thay đổi giá trị này thành

ceil(0.1+0.2)\*10 và bạn nghĩ rằng kết quả này sẽ

trả về 3 vì (0,1+ 0,2=0,3) và (0.3\*10=)3 và khi đó ceil sẽ chỉ cho kết quả là 3 nhưng trên thực tế, kết quả này là 4 và đó là vì

0,1 +0,2 thực sự bằng

một số 0.300000000000

dãy số 0 rồi đến 4 ở cuối

và khi bạn nhân số này với 10

bạn sẽ 3.000000000004 và khi bạn áp ceiling cho

số này, nó làm tròn lên đến bốn

A computer screen shot of a number

Description automatically generated

chứ không phải ba bởi vì bạn có số thập phân cuối cùng ở đây là 4 nếu đây bằng 0 thì số này sẽ không được làm tròn thành 4, nó sẽ vẫn là 3 nên về cơ bản đừng bao giờ tin các số thực đến chữ số cuối cùng và không bao giờ so sánh trực tiếp các số thực để tìm đẳng thức.

Tôi sẽ để lại một liên kết trong phần mô tả nơi bạn có thể đọc thêm về các số thực và cách so sánh chúng cho bằng nhau một cách chính xác. Hãy để tôi chỉ cho bạn ý của tôi bằng cách không so sánh trực tiếp các luồng, giả sử bạn có

x=0,23 và sau đó y=1-0,77

và hãy bỏ những cái này ra chúng ta thấy rằng cả hai đều giống nhau cả hai đều bằng 0,23

A black background with a green light

Description automatically generatednếu chúng ta thực sự so sánh nó thì sao nếu chúng ta làm vậy nếu x bằng y và đừng lo lắng về điều kiện if chúng ta sẽ nói về nó trong một video riêng nhưng nếu chúng ta làm điều này hãy làm( echo=yes) nếu

Không( echo=no ) bạn sẽ nghĩ rằng nó sẽ in ra YES vì chúng

giống nhau về mặt kỹ thuật nhưng nếu làm mới thì chúng ta sẽ nhận được NO

A screen shot of a computer

Description automatically generated

và đó là ý tôi bạn không bao giờ nên so sánh trực tiếp các số float như thế này.

Một điều nữa cần lưu ý là một số thao tác hoặc phép tính có thể dẫn đến một giá trị không xác định, giá trị này sẽ được biểu thị bằng một hằng số có tên là NAN,

vì vậy nếu chúng ta lặp lại một hằng số thì chúng ta sẽ thấy

giá trị đó là NAN.

A black rectangular object with a black border

Description automatically generated with medium confidence Bạn biết NAN là viết tắt của không phải là số và bạn có thể nhận được giá trị này khi một số thao tác hoặc phép tính không thể được tính toán và nó chỉ dẫn đến không phải là số nên một trong những phép tính đó sẽ là log(-1)

và điều này cũng sẽ dẫn đến NAN

A black screen with white text

Description automatically generated

một hằng số khác gọi là vô cực là INF

và nếu chúng ta lặp lại nó, chúng ta sẽ nhận được thông tin này

A black and white photo

Description automatically generated và bạn nhận được thông tin này

khi bạn vượt quá giới hạn float, vì vậy nếu bạn có

PHP\_FLOAT\_MAX\*2

thì bạn sẽ nhận được vô cùng

A black and white photo

Description automatically generated with medium confidencevì số đó không thể thực sự được biểu diễn bởi vì đây là giá trị float tối đa có thể được lưu trữ trên nền tảng và khi bạn nhân số đó với 2, nó sẽ in ở dạng vô cực ở đây nên về cơ bản, bất kỳ phép tính nào đẩy số nổi vượt quá giới hạn cũng

sẽ dẫn đến vô cùng, bạn cũng không bao giờ nên so sánh

trực tiếp một biến với vô cực hoặc với một NAN

để kiểm tra xem biến đó là vô cùng hay không. Thay vào đó, bạn nên sử dụng các hàm dựng sẵn có tên là NAN và INF, chẳng hạn như nếu chúng ta gán cái này cho

x, hãy xóa cái này khỏi đây(ceil(0.1+0.2)\*10) và chúng ta gán var\_dump($x)

chúng tôi thấy rằng nó vẫn là kiểu dữ liệu chảy nhưng đây thực sự không phải là một con số, vậy làm cách nào chúng tôi kiểm tra xem nó có phải là một hàm INF hay không

và chúng tôi nhận được bool true

A black screen with a white stripe

Description automatically generated, điều ngược lại với điều này sẽ là hữu hạn, về cơ bản cho biết nếu số đó không phải là vô hạn thì trong trường hợp này điều này sẽ false

A screen shot of a computer

Description automatically generated nhưng nếu chúng ta đặt x thành giá trị nào đó như 5 thì điều này sẽ

đúng để kiểm tra xem biến đó có phải là số hay không,

A black rectangular object with a white line

Description automatically generated with medium confidence

bạn sẽ sử dụng cái gì đó gọi là is\_nan và điều đó sẽ trả về true hoặc sai

và trong trường hợp này nó sẽ trả về sai,

A screen shot of a computer

Description automatically generated

hãy ghi log(-1) và điều này sẽ trả về đúng ngay bây giờ

A screen shot of a computer

Description automatically generated

và cuối cùng trước khi chúng ta kết

thúc video này, hãy nói về việc truyền(casting), vì vậy giả sử

bạn có một số nguyên như thế này và bạn thực hiện var\_dump

x kiểu dữ liệu là int theo cách bạn có thể chuyển đổi nó thành float

bằng cách sử dụng float và điều này sẽ chỉ chuyển số đó thành float theo những cách khác

A black and white photo

Description automatically generated with medium confidence

bằng cách sử dụng một hàm có tên floatval và về cơ bản nó có

tác dụng tương tự .......

nhưng cá nhân tôi thì không như gọi các hàm không cần thiết vì

thực sự không cần phải gọi hàm này,

bạn có thể chỉ cần thực hiện float như thế này, theo tôi thì tốt hơn có một số điều cần lưu ý khi chuyển đổi các kiểu dữ liệu khác thành float khi chuyển đổi một chuỗi thành float nếu chuỗi có thể được biểu diễn dưới dạng số, nghĩa là nếu nó là số thì nó sẽ được chuyển thành float một cách chính xác, nếu không nó sẽ được chuyển thành 0, vì vậy nếu chúng ta

đặt chuỗi này thành $x=’awdawda’ thì đó không phải là số

và chúng ta chuyển đổi nó thành float, chúng ta sẽ nhận được kiểu dữ liệu float nhưng là số 0

nếu chúng ta có một số con số ở đây như thế này thì con số này sẽ được chuyển đổi thành 15,5

A black and white background

Description automatically generated with medium confidence

Đây là số cho video này trong video tiếp theo,

chúng ta sẽ nói về dây,

cảm ơn bạn rất nhiều vì đã xem, vui lòng

nhấn thích và đăng ký và tôi sẽ gặp lại bạn ở video tiếp theo.