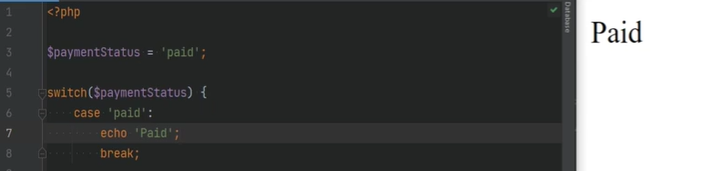
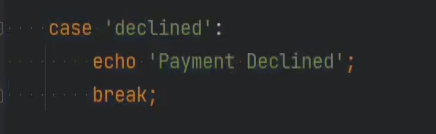
Câu lệnh switch tương tự như câu lệnh if else if và else và nó cho phép bạn so sánh cùng một biểu thức với nhiều giá trị khác nhau và chạy các phần mã khác nhau dựa trên giá trị nào khớp với biểu thức đã cho, vì vậy trong ví dụ này:



ngay tại đây chúng ta viết câu lệnh switch bằng cách sử dụng từ khóa *switch* và trong *dấu ngoặc đơn*, chúng tôi cung cấp biểu thức và dựa trên những gì biểu thức này đánh giá để chúng tôi kiểm tra nhiều trường hợp khác nhau,

biểu thức này có thể phức tạp nếu cần hoặc đơn giản chỉ là một biến trong trường hợp này nó chỉ là một biến được gọi là trạng thái thanh toán và tôi chỉ đang kiểm tra các trạng thái thanh toán khác nhau nên nếu trạng thái thanh toán được thanh toán, tôi muốn thực hiện một số hành động nhất định nếu trạng thái thanh toán bị từ chối, tôi muốn in từ chối thanh toán và sau đó có thể thực hiện một số hành động khác, v.v.

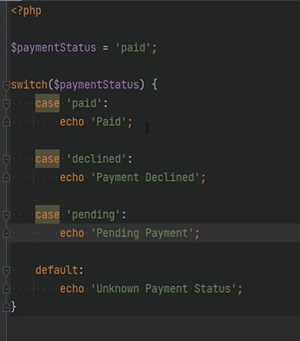
Nếu không tìm thấy kết quả khớp thì trường hợp mặc định sẽ được sử dụng vì vậy nếu không tìm thấy kết quả khớp thì đây là những gì sẽ được thực thi trường hợp mặc định là không bắt buộc nên bạn không cần phải sử dụng nó nếu bạn không cần lưu ý việc sử dụng của



câu lệnh break ở đây, chúng tôi đã giới thiệu câu lệnh break trong video trước khi làm việc với các vòng lặp và nó cũng phục vụ cùng mục đích trong câu lệnh switch để giải thích rõ hơn cách hoạt động của break trong câu lệnh switch,

trước tiên bạn cần biết chính xác câu lệnh switch hoạt động như thế nào. Bản thân cách hoạt động của câu lệnh switch là nó thực thi từng dòng một nhưng nó chỉ bắt đầu thực thi từng dòng sau khi tìm thấy trường hợp khớp,

ví dụ:



các khối này ở đây không bao giờ được thực thi nếu không tìm thấy trường hợp khớp nào khi một trong các trường hợp vượt qua kiểm tra sau đó nó bắt đầu thực thi các câu lệnh theo từng dòng cho đến khi kết thúc khối chuyển đổi hoặc cho đến khi câu lệnh ngắt đầu tiên ở trạng thái hiện tại ở đây chúng ta có trạng thái thanh toán bằng với các trường hợp đã thanh toán và trường hợp đầu tiên khớp và do đó nó bắt đầu thực thi từng dòng nên chúng ta đang lặp lại đã thanh toán hết và sau đó chúng tôi đạt được câu lệnh break ở đây nên nó dừng thực thi

tuy nhiên nếu bây giờ chúng tôi xóa câu lệnh break khỏi đây thì khi trường hợp này trôi qua, nó bắt đầu thực thi lại từng dòng một, nó báo lại đã thanh toán nên không có câu lệnh break nên nó chuyển sang câu lệnh tiếp theo câu lệnh thanh toán echo bị từ chối và nó thực thi câu lệnh đó và việc trường hợp này có vượt qua hay không không quan trọng

vì nó đã vượt qua

trường hợp trước nên ngay khi một trong các

trường hợp trôi qua thì các câu lệnh khác sẽ được

thực thi trừ khi có

câu lệnh break vì ở đó là một

tuyên bố ngắt ở đây, nó cũng sẽ ngừng thực thi

ở đây

nhưng nó sẽ in từ chối thanh toán, vì vậy

nếu chúng tôi làm mới trang, chúng tôi sẽ được thanh toán và

từ chối thanh toán nếu chúng tôi xóa

tuyên bố ngắt khỏi đây thì

bây giờ, tuyên bố ngắt tiếp theo sẽ ở đây và

sau đó bạn sẽ dừng ngay tại đây nếu chúng tôi

xóa câu lệnh break khỏi đây

thì mọi thứ sẽ được thực thi cùng

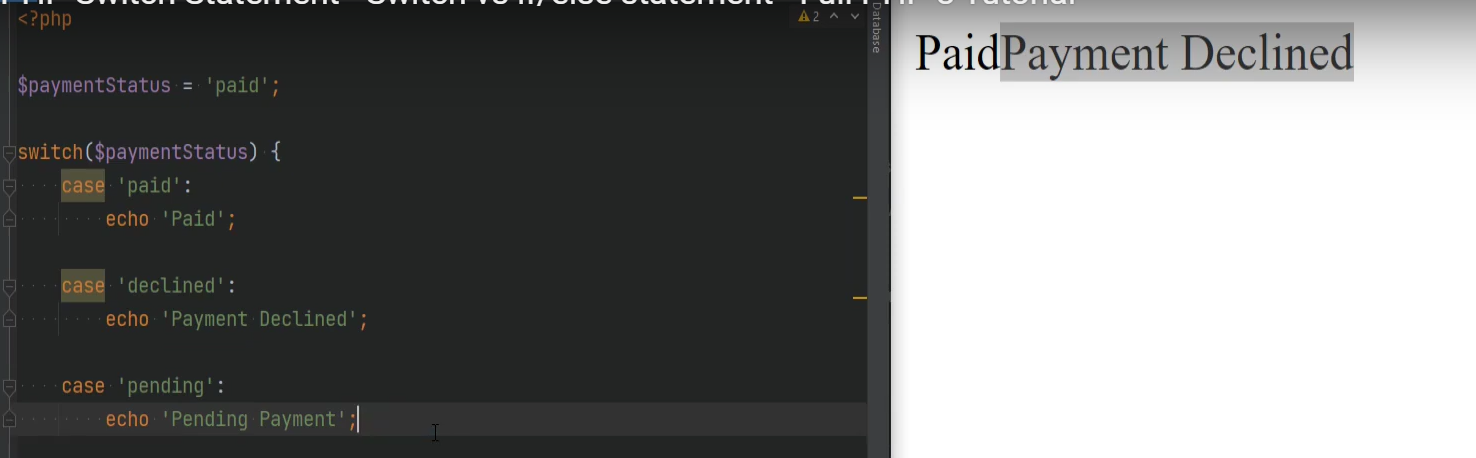
với trường hợp mặc định, vì vậy nếu làm mới,

chúng tôi sẽ thấy rằng mọi thứ được in ra như

bạn có thể thấy đây có thể là

hành vi có chủ ý hoặc không và nó có thể dẫn đến

một số lỗi không mong muốn



mặc dù có một số trường hợp sử dụng cần

thực hiện theo chiến lược mùa thu này

và một trong các trường hợp sử dụng

là bạn có thể muốn thực thi cùng một

khối mã cho nhiều giá trị,

chẳng hạn như giả sử chúng tôi có một

trạng thái khác được gọi là default

, một tùy chọn sẽ

là chỉ sao chép

khối mã này cho khách hàng và chỉ

thay đổi khối mã này thành bị từ chối

và chỉ chạy cùng một khối mã với khối mã

bị từ chối và điều này sẽ hoạt động nhưng nó

được gọi là sao chép đúng và chúng tôi không

muốn sao chép

mã, giải pháp cho điều đó là để xóa

khối mã này khỏi máy khách

và chỉ cần chuyển qua trường hợp khác

bị từ chối ở đây và thêm

câu lệnh ngắt vào đó, điều này về cơ bản

có nghĩa là nếu trạng thái thanh toán bị từ chối

hoặc trạng thái thanh toán bị từ chối thì hãy chạy

khối mã này nếu chúng tôi thay đổi điều này bị

từ chối

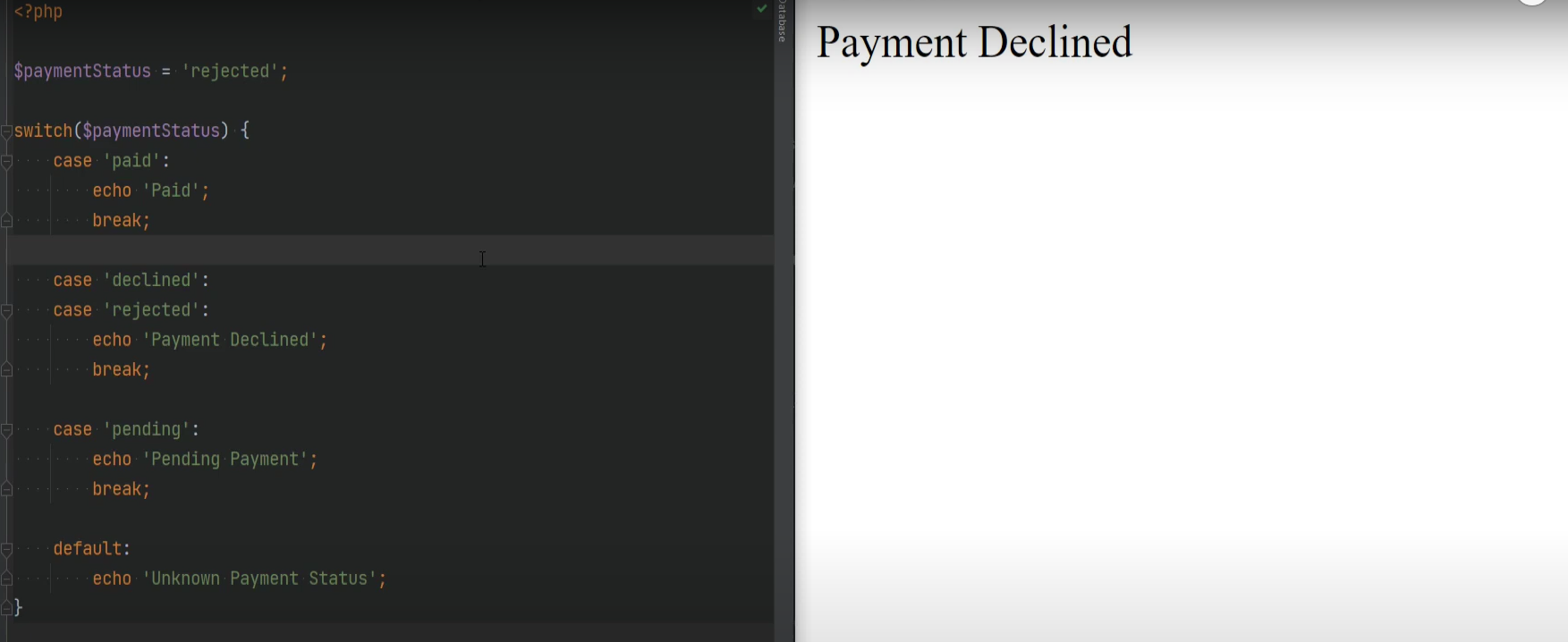
và chúng tôi làm mới trang mà chúng tôi nhận được khoản thanh toán

bị từ chối nếu chúng tôi thay đổi điều này thành bị

từ chối, chúng tôi nhận được cùng một ghi chú đầu ra

rằng câu lệnh chuyển đổi thực hiện

so sánh lỏng lẻo, ví dụ:

 nếu thay vì

biểu diễn chuỗi

các trạng thái thanh toán, hãy nói rằng chúng tôi có

số nguyên biểu diễn hoặc một số loại

biểu diễn bằng số và giả sử

rằng trạng thái thanh toán bằng một

và sau đó trạng thái từ chối bằng hai

bị từ chối là ba và đang chờ xử lý bằng 0

thì ở trạng thái thanh toán, giả sử

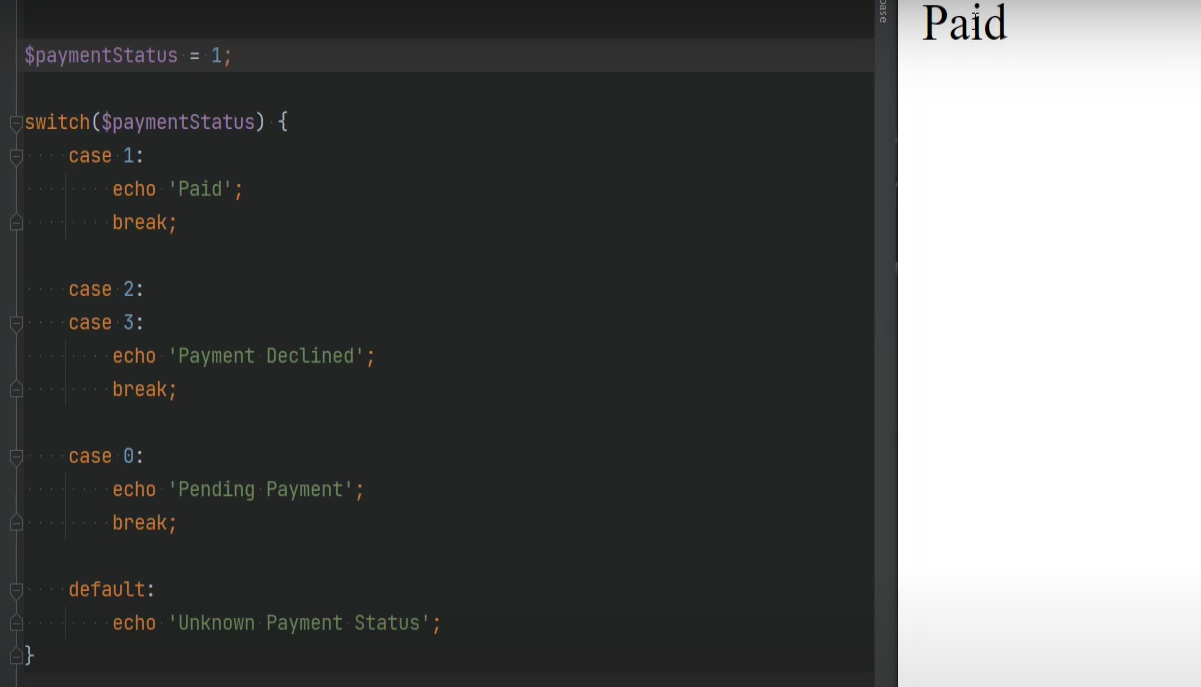
rằng chúng tôi đặt nó thành một, chúng tôi làm mới

mọi thứ vẫn đang hoạt động,

trạng thái thanh toán được thanh toán nếu chúng tôi thay đổi chuỗi này

thành chuỗi một và chúng tôi làm mới đó là hai lần

in đã thanh toán

Điều đó có nghĩa là nó đang

thực hiện so sánh lỏng lẻo và nó không

kiểm tra các loại dữ liệu nên dữ liệu

này vẫn khớp với một chuỗi ngay cả khi chúng tôi

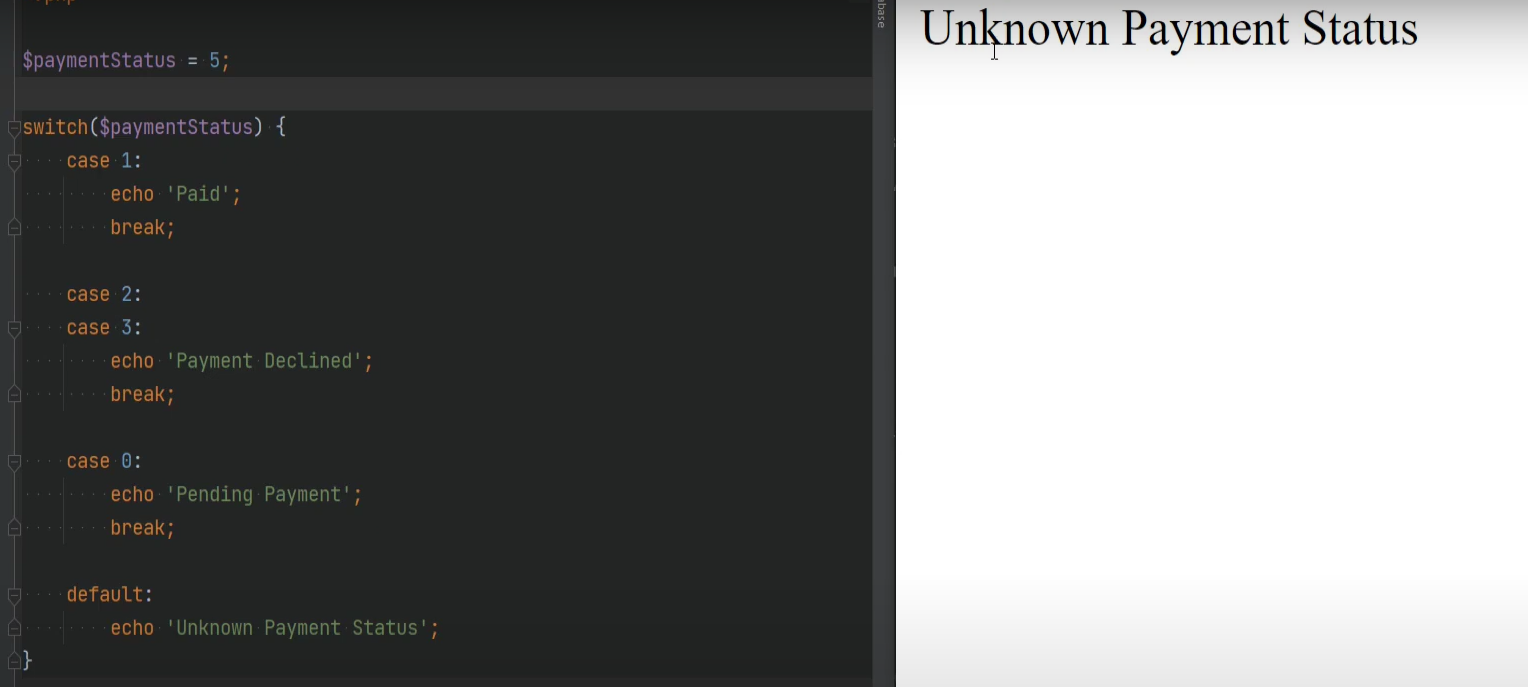
thay đổi điều này thành một cái gì đó giống như

true, điều này vẫn sẽ khớp với một nếu

chúng ta thay đổi nó thành một cái gì đó không

tồn tại

thì chúng ta sẽ nhận được trạng thái thanh toán không xác định.



Một điều khác cần lưu ý là khi

bạn đang sử dụng câu lệnh switch trong

vòng lặp, câu lệnh break ở đây sẽ chỉ

bị hỏng nằm ngoài câu lệnh chuyển đổi này

chứ không nằm ngoài vòng lặp vì vậy hãy để tôi chỉ cho

bạn một ví dụ nhanh. Tôi sẽ đặt

trạng thái thanh toán này thành trạng thái thanh toán

và đặt trạng thái đó thành mảng và giả sử chúng ta

có một loạt trạng thái thanh toán ở đây

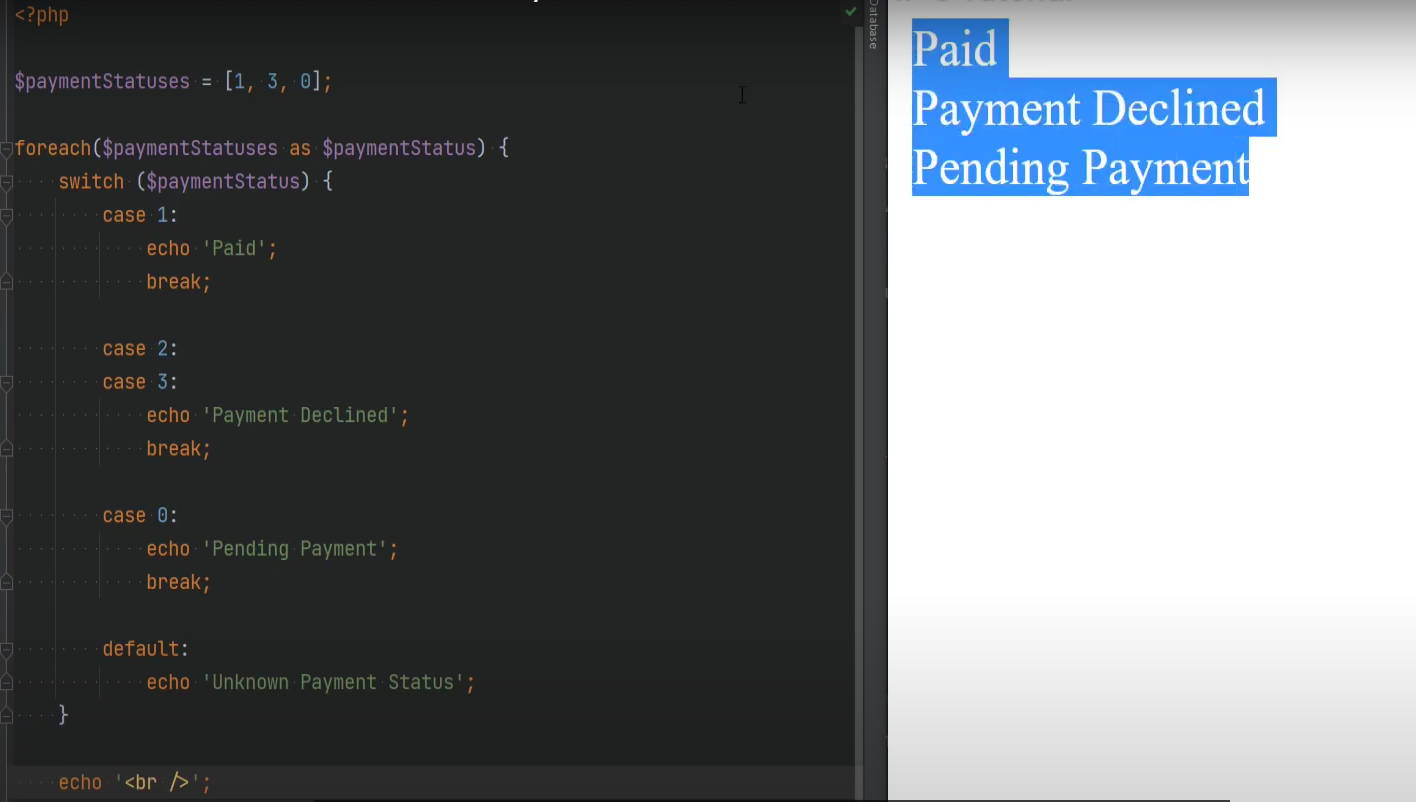
và sau đó chúng ta &#39;sẽ thực hiện cho mỗi

trạng thái thanh toán vòng lặp làm trạng thái thanh toán và

hãy thêm một dòng ngắt ở đây. Chúng

tôi làm mới trang, chúng tôi nhận được ba

trạng thái thanh toán vì chúng tôi có ba phần tử



trong mảng, điều đó có nghĩa là

các câu lệnh ngắt ở đây

chỉ thoát ra khỏi công tắc này

câu lệnh nếu bạn muốn thoát khỏi

vòng lặp vì bất kỳ lý do gì bạn có thể

làm điều đó bằng cách chỉ định đối số tùy chọn

ở đây để cho php biết

bạn muốn thoát ra bao nhiêu cấp độ, vì vậy trong

trường hợp này chúng tôi sẽ chỉ định hai

và nó sẽ thoát ra khỏi chuyển đổi và

cả vòng lặp nếu bạn làm mới trang,

chúng tôi chỉ nhận được văn bản được trả tiền, bạn cũng có thể

sử dụng câu lệnh tiếp tục thay vì

câu lệnh break và nó hoạt động

tương tự như câu lệnh break nhưng nó

sẽ dẫn đến cảnh báo, chẳng hạn như nếu

chúng tôi thay thế câu lệnh break này hai với tiếp tục

và chúng tôi làm mới trang và cảnh báo

ở đây cho bạn biết rằng

tiếp tục hoạt động theo cách tương tự như ngắt và nếu

bạn thực sự muốn tiếp tục vòng lặp

thì bạn cần chỉ định

đối số tùy chọn và trong trường hợp này

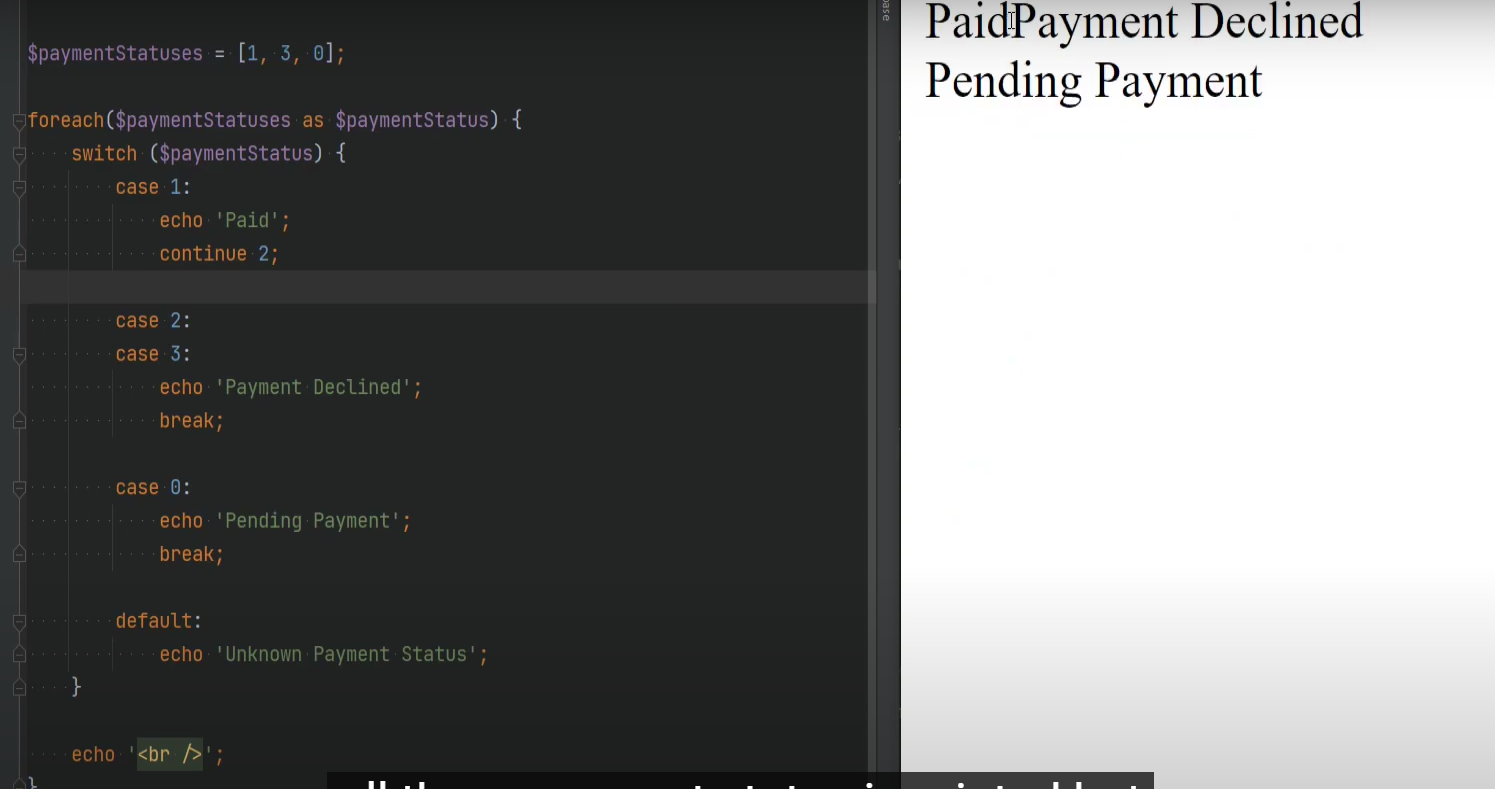
nếu chúng tôi chỉ định đối số ở đây thì nó

sẽ bỏ qua bước lặp cho

vòng lặp nếu chúng ta làm mới trang, chúng ta nhận được

cả ba trạng thái thanh toán được in nhưng

không có dòng ngắt



và đó là vì nó không đến

dòng này ngay tại đây sau khi

trường hợp đầu tiên trôi qua vì chúng ta có

tiếp tục ở đây để đầu tiên nó in

số tiền đã thanh toán như ở đây và sau đó

tiếp tục

nếu chúng tôi chỉnh sửa phần này trước khi in đã thanh toán

thì chúng tôi sẽ không thấy số tiền đã thanh toán. Bạn có thể

hỏi sự khác biệt giữa

chuỗi câu lệnh if và

câu lệnh switch và câu lệnh switch là gì điểm khác biệt chính

giữa hai loại này

là biểu thức ở đây trong

câu lệnh switch được thực thi

một lần trong khi biểu thức bên trong

câu lệnh điều kiện if else if được

thực thi cho mỗi câu lệnh và tôi

sẽ cho bạn xem một ví dụ,

giả sử rằng chúng ta có một số loại

hàm nặng thực hiện một số

thao tác nặng và sau đó có thể mất khoảng

ba bốn giây vì vậy hãy gọi

hàm đó là x chẳng hạn, sau đó chúng ta có thể làm

điều gì đó như thế này

và tôi sẽ định nghĩa hàm

ở đây và đừng lo lắng về các hàm

này Chúng tôi sẽ đề cập đến các chức năng

trong các video sau

và để mô phỏng rằng chức năng này mất

vài giây, chúng tôi sẽ thêm chế độ ngủ

ở đây

và chức năng ngủ ở đây sẽ chỉ

tạm dừng quá trình thực thi tập lệnh trong một số

giây để chúng tôi có thể nói là ngủ trong một vài giây.

3 giây rồi echo xong, bây giờ chúng ta có thể

trả về

1 vì x trả về 1,

điều kiện đầu tiên ở đây sẽ vượt qua

và do đó chúng ta sẽ in một cái để

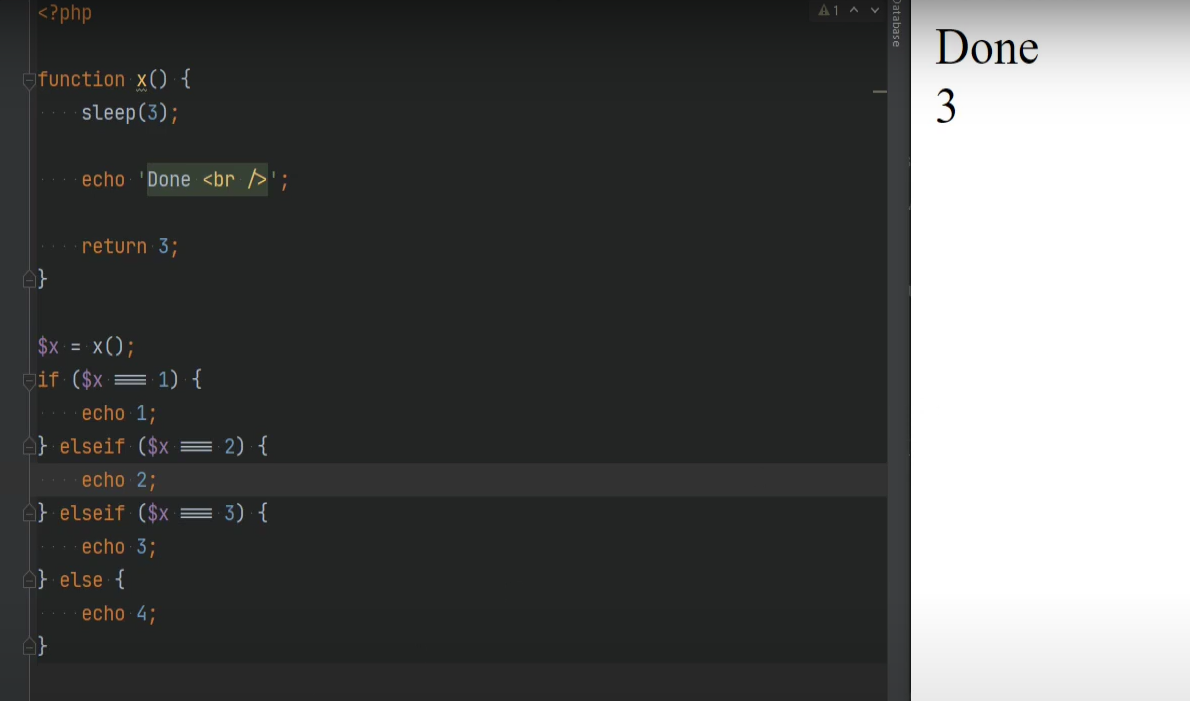
những cái khác không cần chạy nên nếu chúng ta

làm mới thì

nó sẽ đợi ba giây

giây, sau đó nó thực hiện và in

một,



tuy nhiên nếu chúng ta trả về một cái gì đó như

ba thì nó phải kiểm tra cái đầu tiên nếu

có điều kiện ở đây, sau đó anh ta phải kiểm tra

elsif và sau đó anh ta phải kiểm tra một elsif khác

và do đó anh ta chạy hàm x này

ở đây ba lần

anh ấy chạy nó cho mỗi câu lệnh

nên bây giờ nó sẽ ở chế độ ngủ trong chín

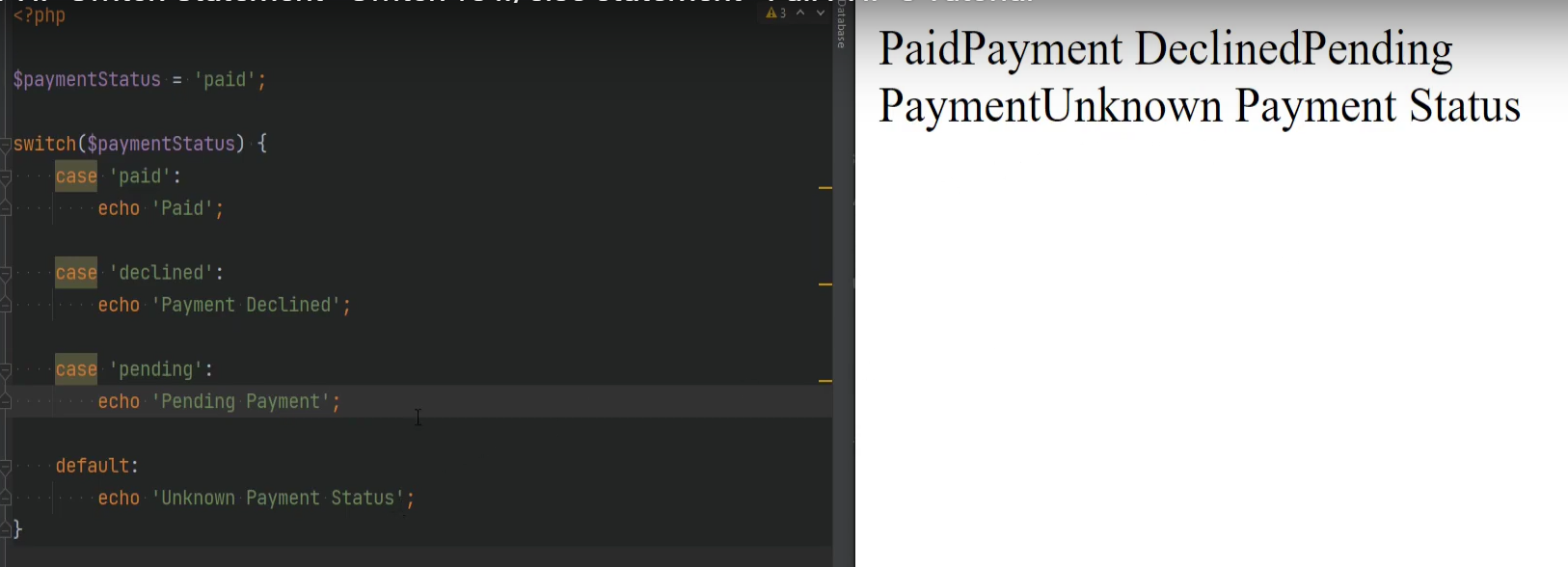
giây vì vậy nếu chúng tôi làm mới như bạn có thể thấy,

nó sẽ mất một lúc và sau chín

giây nó sẽ in

xong và ba giây và bạn thấy rằng nó đang

in ba lần ở đây.



có nghĩa là nó đang thực thi ba lần,

giải pháp cho vấn đề này thực sự

khá đơn giản, bạn có thể chuyển

lệnh gọi hàm sang một biến như thế này

và sau đó chỉ cần sử dụng x ở đây và ngay bây giờ nếu bạn

làm mới thì sẽ chỉ mất

ba giây và nó sẽ chỉ mất ba giây chạy một lần

tuy nhiên trong câu lệnh switch, nó

chỉ thực thi một lần và bạn không cần

gán nó cho một biến, vì vậy nếu chúng ta làm

điều tương tự với câu lệnh switch, chúng ta

sẽ làm một cái gì đó như

switch x và sau đó viết hoa một echo

một và chúng ta&#39; Tôi sẽ làm tương tự cho

các trường hợp khác và nếu chúng tôi làm mới trang thì

sẽ chỉ mất ba giây

và nó sẽ thực thi một lần nên

bạn thấy biểu thức ở đây chỉ

thực hiện một lần và sau đó so sánh

kết quả của nó với các giá trị này.

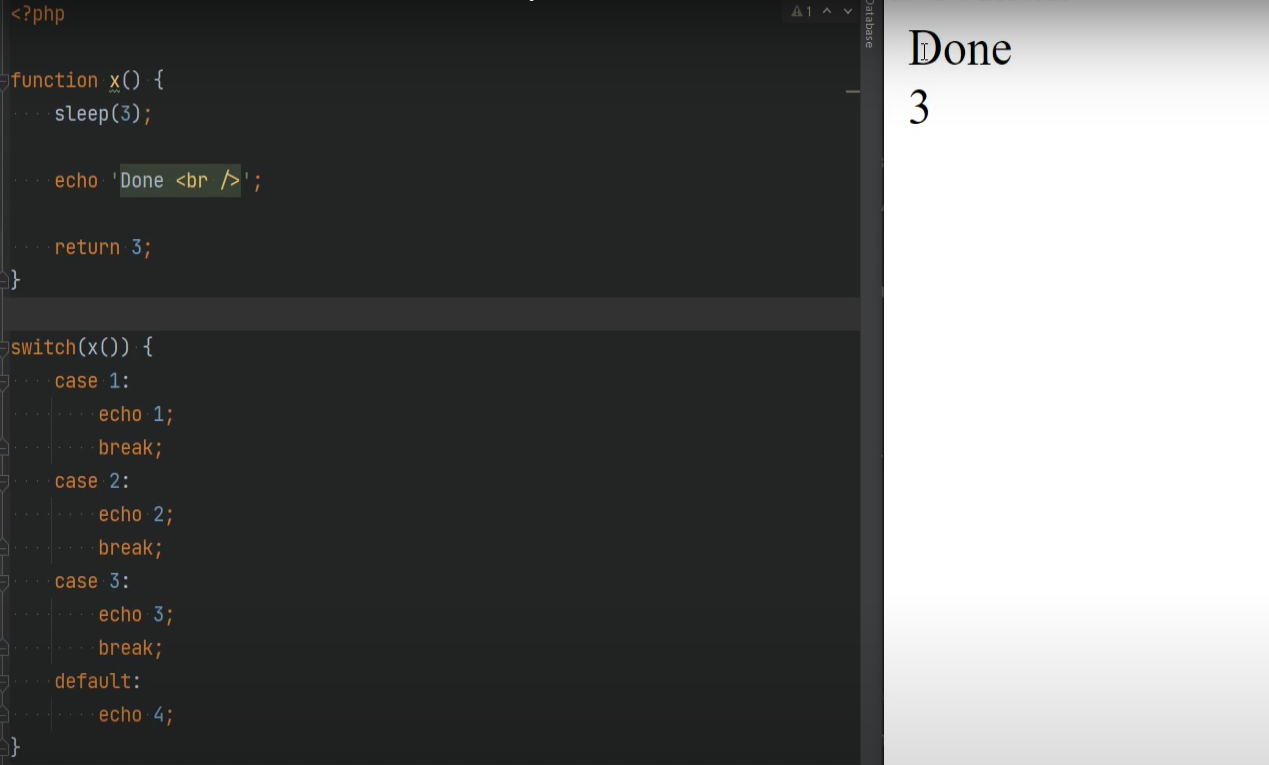
Câu lệnh switch tổng thể nhanh hơn một chút

so với câu lệnh if nhưng đừng

lo lắng về hiệu suất vì nó

không đáng kể nhưng trong những trường hợp như

ví dụ này

 thì bạn sẽ muốn

gán hàm cho một biến

hoặc sử dụng câu lệnh switch, đây là câu lệnh

dành cho việc này video trong video tiếp theo chúng ta

sẽ nói về biểu thức so khớp

được giới thiệu trong php vì vậy cảm ơn bạn

rất nhiều vì đã xem, vui lòng nhấn thích và

đăng ký và tôi sẽ gặp bạn ở video tiếp theo