2151163696 – 63HTTT1

# 1.16. PHP Loops Tutorial - Break & Continue Statements - Full PHP 8 Tutorial

* ­Tương tự như các ngôn ngữ lập trình khác, vòng lặp trong PHP được sử dụng để thực thi các câu lệnh nhiều lần.
* Php hỗ trợ các vòng lặp, có vòng lặp while do, vòng lặp while, vòng lặp for và vòng lặp 4h.
* Với vòng lặp while: một bộ đếm đơn giản và in các chữ số lên tới 15, biến i=0 và sau đó i=<1, lặp lại i và tăng giá trị

A screenshot of a computer

Description automatically generated

* Nếu làm mới trang, nhận được các chữ số từ 0 đến 15 như hình.

A screenshot of a computer

Description automatically generated

* Như vậy có thể thấy cú pháp của vòng lặp while tương tự như câu lệnh điều kiện if .
* Bắt đầu bằng từ khóa while, theo sau là dấu ngoặc đơn chứa biểu thức điều kiện và biểu thức này đánh giá là đúng, sẽ tiếp tục chạy các câu lệnh chứa trong dấu ngoặc nhọn, biểu thức điều kiện này được đánh giá ở đầu mỗi lần lặp và ngay khi nó được đánh giá là sai thì vòng lặp sẽ dừng, không bao giờ chạy.
* Vì vậy nếu đặt i = 20 và làm mới trang, ta nhận được

A screenshot of a computer

Description automatically generated

* Nếu quên tăng biến i ở đây thì gọi là vòng lặp vô hạn và vòng lặp vô hạn là thứ sẽ liên tục chạy các câu lệnh vì biểu thức sẽ luôn đánh giá là đúng .

A computer screen shot

Description automatically generated

* Trong trường hợp này, không sửa đổi biến i và i luôn bằng 0, biểu thức này sẽ luôn đúng và có thể gây hại cho ứng dụng . Một điều gì đó giống như while true đang được sử dụng

A black rectangular object with a black border

Description automatically generated

* Một trong những trường hợp sử dụng cho vòng lặp vô hạn là khi muốn đợi điều gì đó xảy ra và sau đó thoát ra khỏi vòng lặp có điều kiện bằng cách sử dụng câu lệnh break,ví dụ như:

A screenshot of a computer program

Description automatically generated

* Nếu i>= 15 break và cần tăng biến i và nếu làm mới trang, nhận được các chữ số từ 0 đến 14 như hình.

A screenshot of a computer

Description automatically generated

* Câu lệnh break chấp nhận một đối số tùy chọn và mặc định được đặt thành 1 nhưng có thể chuyển một đối số khác, ví dụ 2 thì sẽ thoát ra khỏi nhiều vòng lặp lồng nhau.Trong trường hợp này thì không có ý nghĩa gì vì chỉ có một vòng lặp duy nhất.

A screenshot of a computer program

Description automatically generated

* Ví dụ : i > 10 và phá vỡ 2 , sẽ thoát ra khỏi hai cấp độ vòng lặp .
* Nếu đã làm theo cách khác và chỉ sử dụng đối số mặc định là 1 thì chỉ vòng lặp này sẽ bị phá vỡ và vẫn còn một vòng lặp vô hạn, vì vậy nếu làm mới trang, sẽ chỉ nhận được các số lên tới 10.

A screenshot of a computer

Description automatically generated

* Có thể bỏ qua phép lặp vòng lặp hiện tại bằng cách sử dụng câu lệnh continue .
* Ví dụ: Nếu thay đổi thành i =<15 và xóa câu lệnh break thì Nếu i là số chẵn thì sẽ có một vòng lặp vô hạn vì ban đầu đặt = 0 và 0 là số chẵn và do đó sẽ bỏ qua lần lặp này và chuyển sang lần lặp tiếp theo .

Trong trường hợp này, mặc dù không tăng i vì tăng i xuống bên dưới và tiếp tục bỏ qua mọi thứ bên dưới nó . Vì vậy chỉ chuyển sang lần lặp tiếp theo lại = 0 và sẽ tiếp tục lặp đi lặp lại nên tạo ra một vòng lặp vô hạn.

Để khắc phục điều này, cần tăng i ngay trước câu lệnh if có điều kiện hoặc chỉ có thể tăng i trước,và nếu làm . mới trang bây giờ, chỉ nhận được số lẻ.

A screenshot of a computer

Description automatically generated

* Nếu có thể thêm dấu phẩy để phân biệt giữa các số và tương tự như câu lệnh break, continue cũng chấp nhận một đối số tùy chọn cho phép tiếp tục nhiều vòng lặp lồng nhau.

A screenshot of a computer

Description automatically generated

* PHP cũng có cú pháp thay thế cho vòng lặp while, tương tự như các điều kiện if chủ yếu được sử dụng khi nhúng php vào html thay vì mở dấu ngoặc nhọn mà có thể làm dấu hai chấm và thay vì dấu ngoặc nhọn đóng, có thể kết thúc while bằng dấu chấm phẩy và điều này sẽ hoạt động theo cách tương tự vòng lặp do while

A screenshot of a computer

Description automatically generated

* Tương tự như vòng lặp while ,vòng lặp do-while đảm bảo các câu lệnh trong vòng lặp sẽ thực hiện ít nhất một lần vì biểu thức điều kiện được kiểm tra ở cuối một lần lặp thay vì ở đầu như vòng lặp while .
* Nếu biểu thức điều kiện được đánh giá là sai ở đầu thì vòng lặp sẽ không chạy trong vòng lặp do while. Câu lệnh echo ở đây sẽ thực thi lần đầu tiên và sẽ kiểm tra biểu thức điều kiện và quyết định có cần chạy lại hay không.
* Trong trường hợp này, sẽ nhận được các chữ số từ 0 đến 15.

A screenshot of a computer

Description automatically generated

* Nếu thay đổi giá trị thành 25 thì sẽ đánh giá thành sai nhưng câu lệnh echo vẫn sẽ thực thi một lần,

A screenshot of a computer

Description automatically generated

* Vòng lặp for phức tạp hơn so với vòng lặp while hoặc do while, cần có ba biểu thức được phân tách bằng dấu chấm phẩy:
* Biểu thức đầu tiên đánh giá lần đầu tiên
* Biểu thức thứ hai là biểu thức điều kiện được đánh giá ở đầu mỗi lần lặp
* Biểu thức thứ ba được đánh giá ở cuối mỗi lần lặp để có thể thấy ở phía bên phải.
* Có các chữ số được in từ 0 đến 14 vì ở đây đang tạo một biến i gán 0 thì sẽ kiểm tra xem i <15 hay không và việc tăng i mỗi biểu thức này có thể trống vì điều này sẽ hoạt động tốt về cơ bản giống như while, cũng có thể chứa nhiều biểu thức được phân tách bằng dấu phẩy để có thể làm điều gì đó như in i rồi tăng i để có thể nhận xét echo i này và nếu làm mới trang, sẽ nhận được chính xác cùng một kết quả

A screenshot of a computer

Description automatically generated

* Nếu có nhiều biểu thức trong biểu thức điều kiện thì việc đánh giá xem vòng lặp có tiếp tục hay không sẽ được xác định bởi biểu thức cuối cùng
* Ví dụ: có thể in i rồi dấu phẩy sẽ được xóa ,nếu làm mới trang thông báo rằng trước khi số cuối cùng là 14 và bây giờ số cuối cùng là 15.

A screenshot of a computer

Description automatically generated

* Lý do là vì biểu thức điều kiện được chạy cho mỗi lần lặp và lần lặp cuối cùng nhỏ hơn 15 đang được kiểm tra sau khi in Vòng lặp for cũng có thể được sử dụng để lặp các chuỗi và in ra từng ký tự hoặc lặp qua các mảng.
* Ví dụ có một số văn bản tương đương với hello world thì có thể thực hiện ít hơn độ dài chuỗi văn bản , cộng và in từng ký tự trên dòng riêng của nó .Nếu làm mới trang, sẽ nhận được mỗi ký tự trên một dòng riêng

A screenshot of a computer

Description automatically generated

* Nếu đổi ký tự này thành một mảng thì có thể thay đổi độ dài chuỗi thành count vì count trả về độ dài của một mảng và điều này có thể vẫn giữ nguyên và nếu làm mới, sẽ in ra các phần tử của một mảng,

A screenshot of a computer

Description automatically generated

* Có thể làm điều tương tự bằng cách sử dụng vòng lặp do while hoặc while.
* Điều này có vấn đề về hiệu suất vì các biểu thức điều kiện chạy ở đầu mỗi lần lặp .
* Nên gọi hàm count ở đây cho mỗi lần lặp, trong trường hợp này chỉ gọi bốn lần, nhưng trong trường hợp mảng có thể quá lớn hoặc lệnh gọi hàm có thể tốn kém. Vì vậy, có thể tạo một biến có tên là length và gán biến đó trong biểu thức đầu tiên của vòng lặp để có thể làm độ dài để đếm văn bản và biểu thức đầu tiên chỉ chạy lần đầu tiên nên số đếm sẽ chỉ được gọi một lần .Nếu làm mới, sẽ nhận được điều tương tự

A screenshot of a computer

Description automatically generated

* Một tùy chọn khác là thay vì đặt trong vòng lặp for, có thể chỉ cần đặt phía trên vòng lặp như thế này và nếu làm mới, nhận được cùng một lưu ý đầu ra rằng vấn đề về hiệu suất không chỉ áp dụng cho vòng lặp for mà có thể gặp vấn đề về hiệu suất tương tự khi sử dụng vòng lặp while hoặc do while

A screenshot of a computer

Description automatically generated

* Vòng lặp for each cho mỗi vòng lặp được sử dụng để lặp qua các mảng hoặc đối tượng,
* Ví dụ: tạo một mảng các ngôn ngữ lập trình và in từng ngôn ngữ trên dòng riêng để cú pháp dành cho mỗi ngôn ngữ và trong dấu ngoặc đơn, đặt biểu thức có thể lặp lại và trong trường hợp này chỉ là một biến mảng nên sẽ đặt các ngôn ngữ lập trình sau đó sử dụng as và đó là một loại biến nào đó nên trong trường hợp này hãy gọi là ngôn ngữ và sau đó có thể lặp lại ngôn ngữ và sau đó một dòng mới làm mới trang và nhận từng ngôn ngữ lập trình trên dòng riêng

A screenshot of a computer

Description automatically generated

* Trong mỗi lần lặp, sẽ gán giá trị của phần tử hiện tại trong một mảng cho ngôn ngữ biến đổi, nếu cố gắng sử dụng thứ gì đó không phải là mảng hoặc không phải là một mảng đối tượng sẽ báo lỗi.
* Ví dụ: nếu đặt một số dữ liệu vô hướng ở đây, chẳng hạn như kiểm tra và làm mới, sẽ nhận được cảnh báo

A screenshot of a computer

Description automatically generated

* Có thể truy cập các khóa của mảng bằng cách sử dụng biến khóa và sau đó là dấu bằng và dấu lớn hơn , tên biến ở đây có thể là bất cứ thứ gì, không cần phải là khóa hoặc ngôn ngữ.
* Khi tạo mảng trong trường hợp này thì đây là một mảng được lập chỉ mục nên khóa sẽ chỉ là các chỉ mục số nên đây sẽ là 0 1 2 3 và 4. vì vậy có thể thực hiện khóa và làm mới, nhận được là 0 1 2 3 và 4.

A screen shot of a computer

Description automatically generated

* Trong mỗi phần tử hiện tại của mảng được gán cho biến này và được gán bởi một giá trị tuy nhiên có thể gán giá trị đó bằng tham chiếu bằng cách thêm ký hiệu và về cơ bản điều này chỉ gán giá trị của hiện tại phần tử theo tham chiếu. Nếu sửa đổi phần này, sẽ thực sự sửa đổi mảng ban đầu, có thể đặt ngôn ngữ thành php và có thể in ngôn ngữ lập trình .Nếu không thực hiện bằng tham chiếu mà thực hiện theo giá trị khi làm mới trang thì mảng ban đầu không bị thay đổi.

A screenshot of a computer

Description automatically generated

* Lưu ý ở đây là biến mảng, trong trường hợp này biến ngôn ngữ sẽ vẫn giữ nguyên sau 4 giờ vòng lặp đã hoàn tất, giữ câu lệnh echo này ở đây , nếu làm mới trang, nhận được 0 1 2 3 4 và sau đó cũng bị rỉ sét,biến này không bị phá hủy

A screenshot of a computer

Description automatically generated

* Sau Vòng lặp for each nghĩa là nếu sử dụng biến này ở một nơi khác hoặc mắc lỗi đánh máy và sử dụng biến này thay vì một biến khác thì điều này có thể gây ra một số vấn đề trong mã, đặc biệt là khi làm việc với các tham chiếu, khi đang làm việc với thứ gì đó như thế này, biến này sẽ không bị hủy vì vậy nếu làm điều gì đó như ngôn ngữ tương đương với php ở đây và sau đó in ngôn ngữ lập trình, sẽ thấy mục cuối cùng được đổi thành php thì mục cuối cùng đã bị rỉ sét và đã đổi thành php vì ngôn ngữ này vẫn là tham chiếu đến phần tử cuối cùng trong mảng và trong trường hợp này, nó bị rỉ sét và rỉ sét được đổi thành php, mỗi vòng lặp thường chỉ hủy biến bằng cách sử dụng khởi động và theo cách đó chắc chắn rằng điều này sẽ không ghi đè lên bất kỳ thứ gì trong ngôn ngữ lập trình nữa vì vậy nếu làm mới bây giờ, sẽ nhận được một trường hợp sử dụng tốt khác cho vòng lặp 4h là lặp lại trên các mảng kết hợp

A screenshot of a computer screen

Description automatically generated

* Ví dụ: giả sử có một biến người dùng và có một số thuộc tính ở đây như tên email và kỹ năng, có thể sử dụng vòng lặp for each để lặp lại các thuộc tính đó để có thể thực hiện cho từng người dùng làm giá trị khóa và sau đó có thể lặp lại khóa tất cả về giá trị và một dòng mới, làm mới trang mà đang nhận được tên mà nhận được email nhưng đang nhận được mảng cảnh báo này để chuyển đổi chuỗi

A screenshot of a computer

Description automatically generated

* Giá trị này ở đây là một mảng, để giải quyết vấn đề này bằng cách sử dụng json
* Để có thể mã hóa thành json và in ra theo cách khác, mà chúng ta có thể sử dụng implode nhưng chỉ sử dụng implode nếu đó là một mảng. Vì vậy trước tiên sẽ kiểm tra xem có phải là mảng không, giá trị bằng với giá trị implode và sau đó chỉ cần in nó ra.
* Theo cách trên hoặc có thể chỉ cần lặp lại giá trị mà có thể lồng vào mỗi vòng lặp cho mỗi giá trị là kỹ năng và một loại dấu phân cách nào đó rồi xóa cái đó là có thể cải thiện cái này như thế.Nếu không phải là một mảng, chỉ có thể lặp lại giá trị như hiện tại

A screenshot of a computer

Description automatically generated

* Có một cú pháp thay thế cho vòng lặp for each và for vì vậy thay vì dấu ngoặc nhọn mở ở đây, có thể thực hiện dấu hai chấm. Và thay vì dấu ngoặc nhọn cuối có thể kết thúc cho mỗi vòng lặp và làm mới hoạt động theo cách tương tự .Vòng lặp for cũng tương tự như vậy nên ở đây cũng có thể có cú pháp thay thế bằng cách sử dụng dấu hai chấm và số bốn ở cuối và sẽ hoạt động theo cách tương tự

A screenshot of a computer

Description automatically generated

# 1.17. PHP Switch Statement - Switch vs if/else statement - Full PHP 8 Tutorial

* Câu lệnh switch tương tự như câu lệnh if, else if và else , nó cho phép 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.
* Ví dụ : câu lệnh switch bằng cách sử dụng từ khóa switch và trong dấu ngoặc đơn, cung cấp biểu thức và dựa trên những gì biểu thức này đánh giá để kiểm tra nhiều trường hợp khác nhau. Biểu thức này có thể phức tạp nếu chỉ là một biến, trong trường hợp này chỉ là một biến được gọi là trạng thái thanh toán và chúng ta chỉ đang kiểm tra các trạng thái thanh toán khác nhau. Nếu trạng thái thanh toán được thanh toán ,ta muốn thực hiện một số hành động nhất định. Còn nếu trạng thái thanh toán bị từ chối, ta muốn in từ chối thanh toán và sau đó có thể thực hiện một số hành động khác, v.v.

A screenshot of a computer

Description automatically generated

* Nếu không tìm thấy kết quả khớp thì mặc định sẽ là những gì sẽ được thực thi
* Trường hợp mặc định là không bắt buộc nên không cần phải sử dụng nếu không cần lưu ý việc sử dụng của câu lệnh break ở đây.
* Câu lệnh break 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 cần biết chính xác câu lệnh switch hoạt động như thế nào, cách hoạt động của câu lệnh switch là thực thi từng dòng một nhưng chỉ bắt đầu thực thi từng dòng sau khi tìm thấy trường hợp khớp.
* Ví dụ: Các khối này chỉ được thực thi khi tìm thấy một trường hợp khớp. Khi một trường hợp khớp và bắt đầu thực thi các câu lệnh, việc thực thi sẽ tiếp tục cho đến khi kết thúc khối hoặc gặp câu lệnh ngắt đầu tiên. Nếu có câu lệnh ngắt, việc thực thi dừng lại. Nếu không có câu lệnh ngắt, việc thực thi sẽ tiếp tục và các câu lệnh khác trong khối sẽ được thực thi. Việc trường hợp có vượt qua hay không không quan trọng sau khi đã vượt qua một trường hợp trước đó , nên trừ khi có câu lệnh break vì ở đó là một tuyên bố ngắt, cũng sẽ ngừng thực thi ở đây nhưng sẽ in từ chối thanh toán, vì vậy nếu làm mới trang, sẽ được thanh toán và từ chối thanh toán

A screenshot of a computer

Description automatically generated

* Nếu xóa tuyên bố ngắt khỏi đây bây giờ, thì sẽ dừng ngay.
* Nếu xóa câu lệnh break khỏi đây thì mọi thứ sẽ được thực thi cùng với trường hợp mặc định, vì vậy nếu làm mới, sẽ thấy rằng mọi thứ được in ra như sau:

A screenshot of a computer

Description automatically generated

* Có thể thấy hành vi này có thể có hoặc không chủ ý và có thể gây ra một số lỗi không mong muốn. Một số trường hợp sử dụng yêu cầu thực hiện cùng một khối mã cho nhiều giá trị.
* Giải pháp là sao chép khối mã này cho khách hàng và chỉ thay đổi thành trạng thái bị từ chối. Tuy nhiên, điều này được coi là sao chép không chính xác. Một giải pháp khác là xóa khối mã này khỏi máy khách và chuyển sang một trạng thái khác bị từ chối và thêm câu lệnh ngắt vào đó. Điều này có nghĩa là nếu trạng thái thanh toán bị từ chối, chạy khối mã này. Nếu không, làm mới trang.

A screenshot of a computer

Description automatically generated

* Nhận được khoản thanh toán bị từ chối nếu thay đổi điều này thành bị từ chối, 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 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 đặt nó thành một, làm mới mọi thứ vẫn đang hoạt động.

A screenshot of a computer

Description automatically generated

* Trạng thái thanh toán được thanh toán nếu thay đổi chuỗi này thành chuỗi một và làm mới đó là hai lần in đã thanh toán, điều đó có nghĩa là đang thực hiện so sánh lỏng lẻo và không kiểm tra các loại dữ liệu nên dữ liệu này vẫn khớp với một chuỗi ngay cả khi thay đổi điều này thành một cái gì đó giống như true, điều này vẫn sẽ khớp với một nếu thay đổi nó thành một cái gì đó không tồn tại thì sẽ nhận được trạng thái thanh toán không xác định.

A screenshot of a computer

Description automatically generated

* Một điều cần lưu ý khác là trong trường hợp sử dụng câu lệnh switch trong vòng lặp, câu lệnh break sẽ chỉ thoát khỏi câu lệnh chuyển đổi này mà không thoát khỏi vòng lặp.
* Ví dụ, nếu ta đặt trạng thái thanh toán thành một mảng và có một loạt trạng thái thanh toán, sau đó sử dụng vòng lặp cho mỗi trạng thái, ta có thể thêm một câu lệnh break để thoát khỏi vòng lặp. Khi đó, ta sẽ nhận được ba trạng thái thanh toán tương ứng với ba phần tử trong mảng. Điều này có nghĩa là câu lệnh break ở đây chỉ thoát ra khỏi câu lệnh switch này mà không thoát khỏi vòng lặp.

A screenshot of a computer

Description automatically generated

* Nếu muốn thoát khỏi vòng lặp vì bất kỳ lý do gì, có thể làm điều đó bằng cách chỉ định đối số tùy chọn ở đây để cho php biết muốn thoát ra bao nhiêu cấp độ. Trong trường hợp này sẽ chỉ định hai và sẽ thoát ra khỏi chuyển đổi và cả vòng lặp nếu làm mới trang, chỉ nhận được văn bản được trả tiền

A screenshot of a computer

Description automatically generated

* Có thể sử dụng câu lệnh tiếp tục thay vì câu lệnh break và hoạt động tương tự như câu lệnh break nhưng sẽ dẫn đến cảnh báo, chẳng hạn như nếu thay thế câu lệnh break này và làm mới trang sẽ cảnh báo ở đây cho biết rằng tiếp tục hoạt động theo cách tương tự như ngắt

A screenshot of a computer

Description automatically generated

* Và nếu thực sự muốn tiếp tục vòng lặp thì cần chỉ định đối số tùy chọn và trong trường hợp này nếu chỉ định đối số ở đây thì sẽ bỏ qua bước lặp cho vòng lặp, nếu làm mới trang, nhận được cả ba trạng thái thanh toán được in nhưng không có dòng ngắt

A screenshot of a computer

Description automatically generated

* Đó là vì không đến dòng này ngay đây, sau khi trường hợp đầu tiên trôi qua vì có tiếp tục ở đây ,để đầu tiên in số tiền đã thanh toán như ở đây và sau đó tiếp tục nếu chỉnh sửa phần này trước khi in đã thanh toán thì sẽ không thấy số tiền đã thanh toán.

A screenshot of a computer

Description automatically generated

* Có thể nhắc đến sự khác biệt giữa câu lệnh if và câu lệnh switch. Câu lệnh switch là một cấu trúc điều khiển mà biểu thức được thực thi một lần duy nhất, trong khi câu lệnh if và else if sẽ kiểm tra và thực thi biểu thức cho mỗi câu lệnh.
* Ví dụ, nếu có một hàm nặng mà thực hiện một số thao tác tốn thời gian, ta có thể gọi hàm đó là 'x' và thực hiện các hành động tương ứng. Vì vậy, ta có thể định nghĩa hàm ở đó và không cần lo lắng về các hàm khác.

A screenshot of a computer program

Description automatically generated

* Và để mô phỏng rằng chức năng này mất vài giây, sẽ thêm chế độ ngủ và chức năng ngủ sẽ chỉ tạm dừng quá trình thực thi tập lệnh trong một số giây để có thể nói là ngủ trong một vài giây. 3 giây rồi echo xong, bây giờ có thể trả về 1 vì x trả về 1, điều kiện đầu tiên sẽ vượt qua và do đó sẽ in một cái để những cái khác không cần chạy ,nên nếu làm mới thì sẽ đợi ba giây giây, sau đó thực hiện và in một

A screenshot of a computer

Description automatically generated

* Nếu trả về một cái gì đó như ba thì nó phải kiểm tra cái đầu tiên nếu có điều kiện, sau đó phải kiểm tra elsif và kiểm tra một elsif khác. Do đó chạy hàm x này ở đây ba lần, chạy nó cho mỗi câu lệnh nên bây giờ sẽ ở chế độ ngủ trong chín giây vì vậy nếu làm mới như có thể thấy, sẽ mất một lúc và sau chin giây sẽ in xong và thấy rằng đang in ba lần ở đây.

A screenshot of a computer

Description automatically generated

* Có nghĩa là đang thực thi ba lần,giải pháp cho vấn đề này thực sự khá đơn giản, 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 làm mới thì sẽ chỉ mất ba giây và sẽ chỉ mất ba giây chạy một lần

A screenshot of a computer

Description automatically generated

* Tuy nhiên, trong câu lệnh switch, chỉ thực thi một lần và không cần gán cho một biến. Vì vậy, nếu ta áp dụng cách tương tự với câu lệnh switch, ta có thể làm như sau: switch(x) và sau đó viết các trường hợp tương ứng và câu lệnh echo. Nếu ta làm mới trang, chỉ mất ba giây và chỉ thực thi một lần. Điều này có nghĩa là biểu thức trong câu lệnh switch chỉ được thực hiện một lần và sau đó so sánh kết quả với các giá trị tương ứng.

A screenshot of a computer

Description automatically generated

* Câu lệnh switch tổng thể nhanh hơn một chút so với câu lệnh if nhưng đừng lo về hiệu suất vì nó không đáng kể .Trong những trường hợp như ví dụ này thì nếu muốn gán hàm cho một biến hoặc sử dụng câu lệnh switch.