

BIỂU THỨC CHÍNH QUY TRONG CHUẨN HÓA DỮ LIỆU FORM

*Cuốn tài liệu này sẽ giúp bạn sử dụng biểu thức chính quy để chuẩn hóa dữ liệu
một cách chặt chẽ, tạo sự an toàn cho hệ thống*

Phan Văn Cường

02/2020

Về tác giả

Chào bạn, tôi chúc mừng bạn đã có cuốn ebook tổng hợp về biểu thức chính quy trong chuẩn hóa dữ liệu form của tôi.

Tôi là Phan Văn Cương founder Unitop.vn, trong quá trình xây dựng ứng dụng tôi nhận thấy rằng việc chuẩn hóa dữ liệu form vô cùng quan trọng.

Tuy nhiên trên thực tế tài liệu Biểu thức chính quy để giúp những bạn mới tiếp cận về lập trình thì vô cùng hiếm hoi.

Chính vì vậy tôi đã quyết định biên soạn cuốn tài liệu này giúp đỡ cho học viên chương trình PHP MASTER của tôi cùng những bạn may mắn nhận được tài liệu này miễn phí.

Bạn chỉ cần nỗ lực đọc nó và áp dụng từng bước thì bạn sẽ biết được cách làm cho dữ liệu hệ thống của mình trở nên sạch sẽ hơn và thống nhất.

Có một điều cản trở duy nhất bạn không được sao chép cuốn tài liệu này nếu chưa có sự cho phép của tác giả.

Chúc bạn thành công!

Phan Văn Cương Founder Unitop.vn

<http://unitop.vn>

Mục lục

Về tác giả	2
Bài 1: Biểu thức chính quy - Regular Expressions là gì? Công dụng của nó?	4
Ví dụ biểu thức chính quy đơn giản	4
Bài 2: Biểu diễn miền giá trị mẫu trong biểu thức chính quy	5
Dấu ngoặc vuông [...] trong biểu thức chính quy	5
Dấu gạch ngang (-) trong ngoặc vuông [...] ở biểu thức chính quy	6
Dấu mũ (^) trong biểu thức chính quy	7
Bài 3: Tính lặp của tập ký tự mẫu trong biểu thức chính quy	7
Sử dụng ký tự *	8
Sử dụng ký tự +	8
Sử dụng ký tự ?	8
Sử dụng ký tự ngoặc kép {}	9
Bài 4: Định vị chuỗi ký tự mẫu trong biểu thức chính quy	10
Định vị chuỗi bắt đầu với ký tự ^	10
Định vị chuỗi kết thúc với \$	10
Bài 5: Nhóm và lựa chọn một trong những trong biểu thức chính quy	11
Nhóm biểu thức mô tả với (...)	11
Sử dụng trong biểu diễn biểu thức chính quy	12
Bài 6: Các ký tự viết tắt trong biểu thức chính quy	13
Bài 7: Kiểm tra địa chỉ Email trong PHP bằng biểu thức chính quy	14
Xây dựng biểu thức chính quy cho (Tên đăng nhập)Username	15
Bước 1: Xây dựng biểu thức chính quy cho Username	15
Bài 9: Kiểm tra định dạng Mật khẩu trong PHP bằng biểu thức chính quy	16
Bài 10. Validation form bằng biểu thức chính quy	18
Biểu thức chính quy	18
Kiểm tra dữ liệu bằng biểu thức chính quy	19
Biểu thức chính quy thường dùng	19
Bài 11. Định hướng xây dựng hàm xử lý validation form php	20
Mục đích	20
Xây dựng hàm	20
Tổng kết	24

Bài 1: Biểu thức chính quy - Regular Expressions là gì? Công dụng của nó?

Biểu thức chính quy (regular expression, viết tắt là regexp, regex hay regxp) là một chuỗi miêu tả một bộ các chuỗi khác, theo những quy tắc cú pháp nhất định. Biểu thức chính quy thường được dùng trong các trình biên tập văn bản và các tiện ích tìm kiếm và xử lý văn bản dựa trên các mẫu được quy định.

Trong các ngôn ngữ lập trình cũng được sử dụng biểu thức chính quy một cách rộng rãi để chuẩn hóa dữ liệu đầu vào hoặc tìm kiếm theo một định dạng mẫu cho trước.

Phần mà chúng ta cần học ở đây là đi sâu vào cách viết các chuỗi mẫu biểu thức chính quy để thực hiện các yêu cầu trong thực tế. Một phần ứng dụng của biểu thức chính quy là Validation Form hoặc làm việc với Htaccess trong lập trình website.

Ví dụ biểu thức chính quy đơn giản

`/abc/`

chuỗi ký tự trên là chuỗi mẫu, và chúng ta dùng nó để kiểm tra các mẫu khác có chứa các ký tự abc hay không?

Chúng ta đi xem xét các chuỗi sau:

`abc def`

Chuỗi này hợp lệ với chuỗi mẫu vì chứa chuỗi abc

`abdefgh123`

Chuỗi này thì không đúng với chuỗi mẫu vì không chứa chuỗi abc

Bài 2: Biểu diễn miền giá trị mẫu trong biểu thức chính quy

Ở bài trước chúng ta đã hiểu qua biểu thức chính quy là gì và thông qua một số ví dụ đơn giản cũng đã hiểu sơ qua về ý nghĩa hoạt động của nó.

Tuy nhiên trên thực tế tập dữ liệu mẫu nó vô cùng đa dạng và phức tạp.

Ví dụ yêu cầu của chúng ta cần một mẫu kiểm tra một trường nhập vào là các chữ số nằm trong tập hợp $\{1,2,..9\}$ chúng ta không thể sử dụng cách thông thường là liệt kê tất cả ra và so sánh.

Hôm nay thông qua một số cú pháp trong biểu thức chính quy thì công việc chúng ta sẽ nhẹ nhàng hơn rất nhiều.

Dấu ngoặc vuông [...] trong biểu thức chính quy.

Khi sử dụng dấu ngoặc vuông này biểu thức chính quy biểu thị chuỗi mẫu cho phép chuỗi so sánh chứa 1 ký tự bất kỳ trong các ký tự nằm ở bên trong.

Ví dụ sau biểu diễn chuỗi mẫu chứa các ký tự abc

`/[abc]/`

Khi đấy các chuỗi : a, ad, abc có định dạng trùng với chuỗi mẫu vì nó tồn tại ít nhất 1 ký tự trong tập hợp {a,b,c}

Tiếp theo, một chuỗi có ký tự chứa một trong các chữ số 1, 2, 3 được biểu diễn dưới dạng biểu thức chính quy như sau:

`/[123]/`

Với lập luận như trên ta có các chuỗi sau đúng theo chuỗi mẫu: 1abc, 345cd

Và các chuỗi sau không đúng định dạng: abc456, cd457

Rồi, chắc đến đây các bạn cũng đã hiểu qua phần nào về cách làm việc của biểu thức chính quy. Chúng ta đi tiếp qua luật tiếp theo.

Dấu gạch ngang (-) trong ngoặc vuông ([...]) ở biểu thức chính quy

Ở phần trước khi chúng biểu diễn tập hợp các ký tự theo cách liên kê, tuy nhiên làm như thế nó rất mất thời gian.

Với dấu gạch ngang sẽ cho phép chúng ta tiết kiệm thời gian khi nó tự động tạo cho chúng ta một dãy các ký tự liên tục (gồm các chữ cái và chữ số).

Ví dụ sau biểu thị chuỗi mẫu các ký tự từ 0-9

```
/[0-9]/
```

Cách biểu diễn như trên nó tương đương với

```
/[0123456789]/
```

Thay vì liệt kê ra từng chữ số thì chỉ cần đưa ra chữ số đầu và cuối dãy.

Biểu diễn các chữ cái từ A đến Z ta viết như sau:

```
/[A-Z]/
```

Biểu diễn các chữ cái từ a đến z nhỏ như sau:

```
/[a-z]/
```

Nâng cao hơn, chúng ta biểu diễn tập hợp ký tự mẫu bao gồm

- Các ký tự chữ số từ 0-9
- Các ký tự chữ cái thường a - z
- Các ký tự chữ cái hoa A - Z

Ta biểu diễn thành chuỗi sau

```
/[A-Za-z0-9]/
```

Đấy, khá dễ hiểu đúng không nào? Tuy nhiên cũng có một số trường hợp tập ký tự mẫu của mình muốn loại bỏ đi một số ký tự hoặc một dãy ký tự nào đó chúng ta làm thế nào? Các bạn tiếp tục theo dõi phần sau.

Dấu mũ (^) trong biểu thức chính quy

Dấu mũ (^) để loại bỏ đi một số hoặc một dãy ký tự của tập hợp ký tự mẫu khi nằm trong ngoặc đơn.

Ví dụ tập ký tự mẫu của chúng ta là bất kỳ loại trừ các ký tự {a,b}.

```
/[^ab]/
```

Hoặc tập ký tự mẫu loại trừ các số 0 đến 3.

```
/[^0-3]/
```

Đấy là cách chúng ta loại trừ đi tập hợp các ký tự hoặc dãy ký tự, chúng ta chỉ cần cho dấu ^ vào trước tập hoặc dãy ký tự muốn loại bỏ.

Bài 3: Tính lặp của tập ký tự mẫu trong biểu thức chính quy

Ở bài trước chúng ta mới tạo ra tập ký tự mẫu nằm trong [...] và chuỗi so sánh chỉ cần có ký tự trùng với một trong các ký tự mẫu của biểu thức chính quy là được.

Tuy nhiên trong thực tế ta muốn so sánh chuỗi có số ký tự nhất định nào đó với điều kiện mỗi ký tự thuộc vào tập ký tự mẫu thì thế nào.

Các cách sau đây cho chúng có thể lặp ký tự theo mong muốn

- Sử dụng ký tự *
- Sử dụng ký tự +
- Sử dụng dấu ngoặc nhọn {}

Để hiểu rõ hơn chúng ta đi qua từng phần một

Sử dụng ký tự *

Cách sử dụng thêm ký tự * trước ký tự hoặc tập hợp có 0 hoặc nhiều lần lặp lại

Ví dụ chuỗi mẫu sau là chuỗi có hoặc không chứa ký tự a

`/a*/`

Với mẫu trên các chuỗi sau trùng với biểu thức mẫu: cd, abc, aac

Sử dụng ký tự +

Với ký tự dấu cộng quy định ký tự hoặc tập ký tự phía trước nó được lặp lại 1 hoặc nhiều lần.

Ví dụ biểu thức chính quy sau yêu cầu chúng ta nhập vào chuỗi có ít nhất một ký tự a.

`/a+/`

Ví dụ một số chuỗi sau phù hợp với biểu thức trên: ab, aab

Sử dụng ký tự ?

Đây là cú pháp cho phép ký tự hoặc nhóm ký tự đứng trước nó xuất hiện hoặc không.

Ví dụ biểu thức chính quy sau:

`/a?b/`

Miền giá trị bao gồm các chuỗi b hoặc ab. Dựa theo việc ký tự a có thể xuất hiện hoặc không.

Sử dụng ký tự ngoặc kép {}

Với cú pháp này quy định việc lặp lại của một ký tự hoặc một tập ký tự mẫu.

1. {n} : Trùng khớp khi ký tự hoặc dãy ký tự đứng trước có chính xác n phần tử.

Ví dụ biểu thức sau có tập các trường hợp trùng khớp là các số có 4 chữ số trong đó mỗi chữ số thuộc tập từ 0 đến 9.

```
/^[0-9]{4}$/
```

2. {n, m} : Trùng khớp khi ký tự hoặc dãy ký tự đứng trước có n đến m ký tự

Ví dụ sau biểu diễn biểu thức chính quy của Username với luật sau:

- Các ký tự bao gồm các chữ cái hoa, thường, dấu chấm, dấu gạch dưới.
- Có 6 đến 32 ký tự.

Ta có biểu thức chính quy sau:

```
/^[A-Za-z0-9_\.]{6,32}$/
```

3. {n,} : Trùng khớp khi ký tự hoặc tập ký tự đứng trước có tổng số ký tự từ n trở lên.

Ví dụ biểu thức chính quy sau biểu diễn các số có lớn hơn 5 chữ số:

```
/^[0-9]{5,}$/
```

Bài 4: Định vị chuỗi ký tự mẫu trong biểu thức chính quy

Như đã giới thiệu ở phần đầu, biểu thức chính quy là một chuỗi dùng mô tả một bộ các chuỗi khác. Tất nhiên có nhiều trường hợp chúng ta cần thêm một điều kiện nữa chính là việc mô tả chuỗi kèm theo cả vị trí.

Một ví dụ nhỏ sau để chúng ta có thể hình dung qua: tôi muốn tìm tất cả các chuỗi có bắt đầu bằng chuỗi **Win**.

Với điều kiện này ta có các chuỗi sau thỏa mãn: **Window** , **Winner**.

Còn các chuỗi sau thì không thỏa mãn điều kiện: The **Window**, He is **Winner**.

Vậy chúng ta cùng nghiên cứu các cách quy định vị trí của biểu thức chính quy:

Định vị chuỗi bắt đầu với ký tự ^

Như ví dụ bên trên chúng ta đã quy định chuỗi bắt đầu là **Win** và nó được mô tả như sau.

```
/^Win/
```

Chú ý: Quy tắc này chỉ đúng khi ký tự ^ không thuộc ngoặc vuông [...]. Trường hợp nó thuộc ngoặc vuông chúng ta đã nghiên cứu các bạn xem thêm tại link sau:

<http://unitop.vn/khoa-hoc/bieu-dien-mien-gia-tri-mau-trong-bieu-thuc-chinh-quy-129.html>

Định vị chuỗi kết thúc với \$

Trong thực tế chúng ta muốn chuỗi so sánh phải có ký tự kết thúc với biểu thức mẫu. Trong biểu thức chính quy có ký tự \$ cho phép chúng ta quy định việc đó.

Ví dụ để mô tả các chuỗi có kết thúc là các chữ cái in thường [a-z] gồm 3 ký tự

```
/[a-z]{3}/
```

Định vị chuỗi bắt đầu và kết thúc `^...$`

Trong đa số trường hợp áp dụng vào lập trình thì chúng ta sử dụng biểu thức chính quy để mô tả các chuỗi có định dạng cho trước và tất nhiên là chuỗi đấy phải khớp 100% với biểu thức mô tả mẫu nên chúng ta thường sử dụng cặp ký tự `^...$`

Ví dụ sau đây để mô tả Username có các ký tự từ A đến Z, a đến z, 0-9, gạch dưới, dấu chấm, có 6 đến 32 ký tự

```
/^[A-Za-z0-9_\.\.]{6,32}$/
```

Bài 5: Nhóm và lựa chọn một trong những trong biểu thức chính quy

Nhóm hay còn gọi là kết nối các phần của biểu thức mẫu là một phần không thể thiếu trong biểu thức chính quy, có giúp chúng ta quản lý chuỗi mô tả một cách rành mạch hơn thông qua việc nhóm từng phần của chuỗi mô tả.

Nhóm biểu thức mô tả với (...)

Sử dụng cặp ký tự ngoặc nhọn giúp chúng ta có thể nhóm các phần của biểu thức mẫu tùy ý.

Ví dụ biểu thức chính quy sau đây mô tả đường link thân thiện website.

```
http://unitop.vn/khoa-hoc/nhom-va-lua-chon-hoac-trong-bieu-thuc-chinh-quy-144.html
```

Phần đầu chính là domain nên ở chúng ta cần mô tả cho chuỗi sau:

```
khoa-hoc/nhom-va-lua-chon-hoac-trong-bieu-thuc-chinh-quy-144.html
```

Cấu trúc link

- Link bắt đầu từ chuỗi 'khoa-hoc'.
- Tiếp theo dấu '/'.
- Tiếp theo là chuỗi các ký tự mô tả của đường link: Bao gồm các ký tự [A-Za-z0-9].
- Tiếp theo dấu '-'.
- Tiếp theo là ID: Bao gồm các ký tự [0-9].
- Tiếp theo là chuỗi '.html'.

Vậy biểu thức mô tả chính quy như sau:

```
^khoa-hoc/([a-zA-Z0-9_-]+)-([0-9]+).html$
```

Vậy với việc sử dụng dấu ngoặc () ta đã tách biểu thức chính quy thành các nhóm con. Phần nào ra phần đó theo đúng tính chất của đường link website như được demo ở trên.

Sử dụng | trong biểu diễn biểu thức chính quy

Kết hợp với cặp dấu ngoặc nhọn () việc sử dụng ký tự | cho ta lựa hoặc phần này hoặc phần khác của tập danh sách các phần tử liệt kê

Ví dụ

```
/gr(a|e)y/
```

Với biểu thức mô tả trên dùng để mô tả một trong hai chuỗi gray hoặc grey chuỗi mô tả của chúng ta được cấu thành từ chuỗi 'gr' tiếp theo nối với (1 trong hai ký tự a hoặc e) và cuối cùng nối với ký tự 'y'.

Ngoài việc lựa chọn một trong tập các ký tự dấu | còn cho phép chúng ta lựa chọn một trong những nhóm con.

```
((4\.[0-3])|(2\.[0-3]))
```

Với chuỗi biểu thức trên mô tả các chuỗi 4.1, 4.1, 4.2, 4.3 hoặc 2.0, 2.1, 2.2, 2.3 .

Bài 6: Các ký tự viết tắt trong biểu thức chính quy

Dưới đây là danh sách các quy tắc viết tắt của biểu thức chính quy phải nắm để nâng cao tốc độ làm việc. Mỗi quy tắc có phần ký tự và mô tả kèm ví dụ của quy tắc đó.

Cách viết	Mô tả
<code>\d</code>	Đại diện cho bất kỳ ký tự [0-9]
<code>\D</code>	Đại diện cho các ký tự không nằm trong [0-9]
<code>\s</code>	Đại diện bất kỳ ký tự trắng
<code>\S</code>	Tập không chứa bất kỳ ký tự trắng
<code>\w</code>	Đại diện cho bất kỳ các ký tự [A-Z], [a-z], [0-9]
<code>\W</code>	Đại diện bất kỳ các ký tự không thuộc dãy ký tự [A-Z], [a-z], [0-9]

Bài 7: Kiểm tra địa chỉ Email trong PHP bằng biểu thức chính quy

Như đã giới thiệu ở trên, sau đây là cách chúng ta kiểm tra xem chuỗi người dùng đúng định dạng email không?

Bước 1

Xây dựng biểu thức chính quy của Email

Email chúng ta có đặc điểm sau:

- Chứa các ký tự từ A đến Z, a đến z
- Các ký tự số
- Ký tự gạch dưới
- Ký tự @
- Các ký tự trước @ có 6-32 ký tự
- Chuỗi ký tự sau @ chia thành hai phần của domain mỗi phần có 2-12 ký tự

Ta có chuỗi biểu thức chính quy theo yêu cầu trên

```
^[A-Za-z0-9_]{6,32}@([a-zA-Z0-9]{2,12})(\.[a-zA-Z]{2,12})+$
```

Bước 2

Kiểm tra bằng hàm `preg_match()`

```
<?php
$partten =
"/^[A-Za-z0-9_\.\.]{6,32}@([a-zA-Z0-9]{2,12})(\.[a-zA-Z]{2,12})+$/";
$subject = "phancuong.qt@gmail.com";
if(!preg_match($partten , $subject))
    echo "Mail bạn vừa nhập không đúng định dạng ";
?>
```

Bài 8: Kiểm tra định dạng Username trong PHP bằng biểu thức chính quy

Tên đăng nhập trong website cũng rất quan trọng, để yêu cầu người dùng đăng ký tài khoản trên website với một định dạng mẫu cho trước tránh trường hợp bị lỗi hổng bị hacker tấn công.

Xây dựng biểu thức chính quy cho Username(tên đăng nhập)

Bước 1: Xây dựng biểu thức chính quy cho Username

Cấu trúc của Username

- Chứa các ký tự A đến Z, a đến z, 0-9 dấu . và dấu gạch dưới
- Độ dài 6 đến 32 ký tự

Ta có biểu thức sau

```
/^[A-Za-z0-9_\.\.]{6,32}$/
```

Bước 2: Kiểm tra chuỗi cho trước với biểu thức chính quy bằng php

Chúng ta sử dụng hàm preg_match() để kiểm tra

```
<?php
$partten = "/^[A-Za-z0-9_\.]{6,32}$/";
$subject = "hocweb123_hello";
if(!preg_match($partten,$subject,$matches))
    echo "Username bạn vừa nhập không đúng định dạng ";
?>
```

Bài 9: Kiểm tra định dạng Mật khẩu trong PHP bằng biểu thức chính quy

Mỗi chúng ta chắc ai cũng đã từng trải nghiệm việc đăng ký tài khoản qua các hệ thống khác nhau như

- Google
- Facebook
- Yahoo
- Twitter
- Skype
- Zalo

-... và rất nhiều hệ thống khác

Những hệ thống trên là những hệ thống lớn việc bảo mật hệ thống là vấn đề luôn được nghiên cứu kỹ, nếu bảo mật không tốt thì thông tin khách hàng bị lộ kéo theo hàng loạt rắc rối về thông tin riêng tư, tài chính...bao nhiêu việc đau đầu kiện tụng đều có thể xảy ra.

Trong hệ thống website thương mại điện tử ở Việt nam cũng vậy cũng rất cần phải được bảo mật tốt để hoạt động mua bán được diễn ra suôn sẻ.

Có một điểm chung tất cả các hệ thống này đều xây dựng riêng cho mình những quy tắc đặt mật khẩu mà họ cho là bảo mật và vừa tiện cho người dùng.

Ví dụ mật khẩu bao gồm

- Các ký tự chữ cái hoa, thường, chữ số, ký tự đặc biệt, dấu chấm.
- Bắt đầu với ký tự in hoa.
- Có từ 6 đến 32 ký tự.

Vậy sau đây chúng ta cùng nhau nghiên cứu việc sử dụng biểu thức chính quy như thế nào.

Bước 1: Xây dựng biểu thức chính quy cho mật khẩu

Như một số yêu cầu về mật khẩu ở trên ta có biểu thức chính quy sau:

```
^([A-Z]){1}([\w_\.!@#$%^&*() ]+){5,31}$
```

Trong đó chuỗi biểu thức chính quy chia thành 2 nhóm

Nhóm 1: Biểu diễn ký tự đầu tiên viết hoa

```
([A-Z]){1}
```

Ở nhóm này ta thấy tập ký tự mô tả của ta từ A-Z đây chính là các chữ cái in hoa và theo yêu cầu của đầu đề thì nhóm này có 1 phần tử.

Nhóm 2: Biểu diễn các ký tự tiếp theo là các chữ cái, chữ số và ký tự đặc biệt

```
([\w_-.!\@#$$%^&*() ]+){5,31}
```

Ở nhóm này ta sử dụng viết tắt trong biểu thức chính quy \w biểu diễn tập các chữ cái và chữ số.

Ngoài ra, để biểu diễn các ký tự đặc biệt ta cần thêm dấu \ vào trước dãy ký tự đặc biệt

```
!\@#$$%^&*()
```

Và như phân tích ở trên thì phần còn lại của chuỗi mô tả mật khẩu còn 5-31 ký tự.

Bước 2: Kiểm tra định dạng mật khẩu bằng preg_match trong php

```
<?php
$partten = "/^([A-Z]){1}([\w_-.!\@#$$%^&*() ]+){5,31}$/";
$subject = "Cuongsb$";
if(!preg_match($partten , $subject, $matchs))
    echo "Mật khẩu bạn vừa nhập không đúng định dạng ";
?>
```

Bài 10. Validation form bằng biểu thức chính quy

Biểu thức chính quy

- Biểu thức chính quy là chuỗi miêu tả một bộ các chuỗi khác, theo những quy tắc, cú pháp nhất định
- Dùng chuẩn hóa dữ liệu đầu vào hoặc tìm kiếm theo một định dạng mẫu cho trước.

Ví dụ:

```
^([A-Z]){1}([\w_\.\!@#$$%^&*() ]+){5,31}$
```

Kiểm tra dữ liệu bằng biểu thức chính quy

- Sử dụng hàm preg_match để kiểm tra khớp dữ liệu.
- Thông qua việc kiểm tra để kết luận và thông báo cho người dùng nếu dữ liệu đầu vào không đúng định dạng mình quy định trước.

```
$pattern = "/^([A-Z]){1}([\w_\.\!@#$$%^&*() ]+){5,31}$/";  
$subject = "Cuongsb";  
if(!preg_match($pattern,$subject))  
    echo "Mật khẩu bạn vừa nhập không đúng định dạng ";
```

Biểu thức chính quy thường dùng

1. Kiểm tra định dạng username

- Username bao gồm các ký tự chữ cái, chữ số, dấu gạch dưới, dấu chấm
- Độ dài 6-32 ký tự

```
/^[A-Za-z0-9_\.\.]{6,32}$/
```

2. Kiểm tra định dạng password

- Mật khẩu bao gồm các ký tự chữ cái, chữ số, ký tự đặc biệt, dấu chấm
- Bắt đầu bằng ký tự in hoa
- Độ dài 6-32 ký tự

```
/^([A-Z])([\w_\.\!@#$$%^&*() ]+){5,31}$/
```

3. Kiểm tra định dạng Email

- Chứa các ký tự chữ cái, chữ số, dấu chấm, gạch dưới.
- Ký tự @.
- Nhóm ký tự trước @ có 6-32 ký tự.
- Nhóm ký tự sau @ là domain một hoặc nhiều cấp.

```
^[A-Za-z0-9_\.\.]{6,32}@([a-zA-Z0-9]{2,12})(\.[a-zA-Z]{2,12})+$
```

Bài 11. Định hướng xây dựng hàm xử lý validation form php

Mục đích

- Định hướng xây dựng các hàm phục vụ cho Validation
- Tăng tính rõ ràng, dễ quản lý cho hệ thống
- Có thể tái sử dụng cho các hệ thống khác nhau

Xây dựng hàm

- is_username() : Kiểm tra định dạng username
- is_password(): Kiểm tra dữ liệu password
- is_email() : Kiểm tra định dạng email
- set_value() : Thiết lập giá trị cho phần tử form
- form_error() : Hiển thị lỗi form

1. Hàm is_username()

```
//@Hàm : is_username
//@Tham số: $username cần kiểm tra
//@Trả về: True nếu đúng định dạng username
```

```
function is_username($username) {
    $parttern = "/^[A-Za-z0-9_\.]{6,32}$/";
    if (preg_match($parttern, $username))
        return true;
}
```

2. Hàm is_password()

```
//@Hàm : is_password
//@Tham số: chuỗi password cần kiểm tra
//@Trả về: True nếu đúng định dạng password
function is_password($password) {
    $parttern = "/^([A-Z]){1}([\w_\.!@#$$%^&*()]){5,31}$/";
    if (preg_match($parttern, $password))
        return true;
}
```

3. Hàm is_email()

```
function is_email($password) {
    $parttern =
"/^[A-Za-z0-9_.]{6,32}@([a-zA-Z0-9]{2,12})(.[a-zA-Z]{2,12})+$/";
    if (preg_match($parttern, $password))
        return true;
}
```

4. Hàm set_value()

```
function set_value($label_field) {
    global $$label_field;
    if (isset($$label_field))
        echo $$label_field;
}
```

5. Hàm form_error()

```
function form_error($label_field) {
    global $error;
    if (isset($error[$label_field])) {
        echo "<span style=\"color:
red;\">>{$error[$label_field]}</span><br/>";
    }
}
```

Code đầy đủ

```
<?php
require 'function/validation.php';
/*
 * BIỂU THỨC CHÍNH QUY
 * 1.Username : /^[A-Za-z0-9_\.]{6,32}$/
 * 2.Password : /^([A-Z]){1}([\w_\.!@#$$%^&*()]){5,31}$/
 * 3.Email:
/^[A-Za-z0-9_.]{6,32}@([a-zA-Z0-9]{2,12})(.[a-zA-Z]{2,12})+$/
*/

if ($_SERVER['REQUEST_METHOD'] == "POST") {
```

```

//Phát cò
$error = array(); // Chưa có lỗi
// Kiểm tra lỗi username
if (empty($_POST['username'])) {
    $error['username'] = "Bạn cần nhập username";
} else {
    // Kiểm tra định dạng username
    if (!is_username($_POST['username'])) {
        $error['username'] = "Username không đúng định dạng";
    } else {
        $username = $_POST['username'];
    }
}
//Kiểm tra lỗi password
if (empty($_POST['password'])) {
    $error['password'] = "Bạn cần nhập vào password";
} else {
    // Kiểm tra định dạng password
    if (!is_password($_POST['password'])) {
        $error['password'] = "Password không đúng định dạng";
    } else {
        $password = $_POST['password'];
    }
}
// Kết luận
if (empty($error)) {
    echo "{$username} <br/> {$password}";
    //Xử lý theo tình huống dữ liệu đã nhập đầy đủ
}
}
?>

<html>
    <head>
        <title>Validation bằng biểu thức chính quy</title>
        <meta charset="utf-8">

```

```

    </head>
    <body>
        <h1>FORM ĐĂNG NHẬP</h1>
        <form action="" method="POST" >
            <label for="username">Tên đăng nhập</label> <br/>
            <input type="text" name="username" id="username"
value="<?php set_value('username'); ?>" > <br/>
            <?php form_error('username'); ?>
            <label for="password">Mật khẩu</label><br/>
            <input type="password" name="password" id="password"
><br/>
            <?php form_error('password'); ?>
            <br/>
            <input type="submit" name="submit_login" value="Đăng
nhập">
        </form>
    </body>
</html>

```

Tổng kết

Vậy là thông qua những bài học vừa rồi tôi đã hướng dẫn đến bạn từng bước các khái niệm về Biểu thức chính quy và cách thức để tạo nên một biểu thức chính quy hoàn chỉnh

Bạn nhớ hãy ghi chép kết hợp với thực hành áp dụng vào thực tế để nắm rõ nội dung hơn.

Chuẩn hóa dữ liệu form là một phần vô cùng quan trọng trong chương trình đào tạo chuyên sâu PHP MASTER.

Chúc bạn thành công!

Phan Văn Cường Founder Unitop.vn

<http://unitop.vn>