

Biểu thức chính quy

RegEx (*regular expression*) là tập hợp những quy tắc, dựa vào những quy tắc này ta sẽ viết ra những biểu thức so khớp giữa các chuỗi với nhau. Đây là một thư viện nâng cao được tích hợp sẵn trong đối tượng RegExp của Javascript.

Trong Javascript thì regular expression là một chuỗi, nhưng nó không được bao quanh bởi cặp dấu nháy đơn ' hoặc nháy kép " mà nó được bao quanh bởi cặp dấu /.

Biểu thức chính quy

Cú pháp:

/pattern/modifiers

pattern là chuỗi Regular Expression

modifiers là thông số cấu hình cho chuỗi pattern, và nó có các giá trị sau:

i : so khớp không quan tâm đến chữ hoa chữ thường

g : so khớp toàn bộ chuỗi cần tìm

m : so khớp luôn cả các dữ liệu xuống dòng (multiline)

Biểu thức chính quy

2. Hàm test() trong javascript RegEx:

Để kiểm thử một biểu thức regular expression thì chúng ta sẽ sử dụng hàm test. Cú pháp của nó như sau:

```
/pattern/.test(string);
```

Trong đó pattern chính là biểu thức chính quy, string chính là chuỗi cần tìm.

Giá trị của hàm này sẽ trả về true nếu tìm thấy, và false nếu không tìm thấy.

Ví dụ: Kiểm tra chuỗi xem trong chuỗi "chào mừng bạn đến với iuh" có xuất hiện chữ "iuh" không thì ta viết như sau.

Biểu thức chính quy

```
let result = /iuh/.test("chào mừng bạn đến với iuh");  
alert(result); // True
```

3. Ký tự bắt đầu và kết thúc chuỗi RegEx trong javascript

Cấu trúc của một chuỗi luôn có ký tự bắt đầu và ký tự kết thúc. Và trong RegEx cũng có một số ký tự giúp quy ước nơi bắt đầu chuỗi và nơi kết thúc chuỗi. Cấu trúc:

```
/^pattern$/
```

Biểu thức chính quy

Ví dụ:

```
<script>
```

```
let pattern = /IUH/i
```

```
let chuoi = "chào mừng bạn đến với iuh"
```

```
let soKhop = pattern.test(chuoi);
```

```
alert(soKhop);
```

```
</script>
```

Biểu thức chính quy

Trong đó ký tự ^ là khai báo bắt đầu chuỗi, còn \$ là khai báo kết thúc chuỗi.

Ký tự bắt đầu chuỗi pattern

```
let result = /iuh/.test("Website iuh");
```

```
Console.log(result); // True
```

Nhưng:

```
let result = /^iuh/.test("Website iuh");
```

```
console.log(result); // False
```

4. Bảng các biểu thức RegEx trong js căn bản

Trước khi làm việc với những ví dụ thì hãy tham khảo bảng tóm tắt dưới đây. Đây là những biểu thức chính quy regular expression căn bản nhất, được sử dụng nhiều nhất trong lập trình javascript.

Biểu thức	Ý nghĩa
[abc]	Tìm các chữ cái a,b hoặc c
[0-9] hoặc [a-z]	Tìm các ký tự từ 0-9 hoặc từ a-z
(x y)	ìm ký tự x hoặc y
\d	Tìm các chữ số
\s	Tìm khoảng trắng
n+	Tìm 1 hoặc nhiều chữ n liên tiếp nhau
n*	Tìm 0 hoặc nhiều chữ n liên tiếp nhau
n?	Tìm 0 hoặc 1 chữ n

Metacharacters

Bảng quy tắc các ký tự đơn.


Expression	Description
.	Tìm ký tự bất kì
\w	Tìm ký tự chữ cái
\W	Tìm các ký tự không phải là chữ cái
\d	Tìm ký tự là chữ số
\D	Tìm ký tự không phải là chữ số
\s	Tìm ký tự là khoảng trắng
\S	Tìm ký tự không phải khoảng trắng
\b	Tìm so khớp bắt đầu hoặc kết thúc chuỗi
\B	Tìm so khớp không phải bắt đầu hoặc kết thúc chuỗi
\0	Tìm ký tự NULL
\n	Tìm ký tự xuống hàng
\t	Tìm ký tự tab

Biểu thức chính quy

Quantifiers

Bảng quy tắc đến số lượng các ký tự.

Expression	Description
+	Kiểm tra ký tự xuất hiện một hoặc nhiều lần
*	Kiểm tra ký tự xuất hiện không hoặc nhiều lần
?	Kiểm tra ký tự xuất hiện không hoặc một lần
{X}	Kiểm tra ký tự xuất hiện đúng X lần
{X,Y}	Kiểm tra ký tự xuất hiện tối thiểu X lần và tối đa Y lần
{X,}	Kiểm tra ký tự xuất hiện ít nhất X lần
^	Kiểm tra ký tự bắt đầu chuỗi
\$	Kiểm tra ký tự kết thúc chuỗi



Biểu thức chính quy

Bài 4: Kiểm tra chuỗi định dạng mã thẻ cào xxxx-xxxx-xxxx-xxxx và x chính là các chữ số.

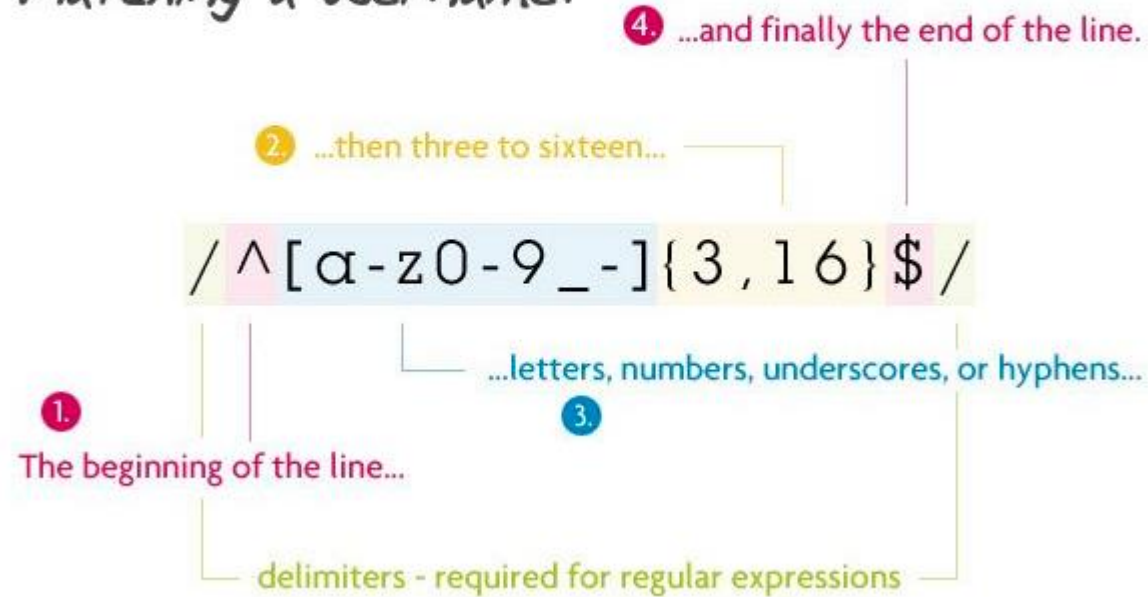
Ta sử dụng [0-9] để kiểm tra là chữ số. Và các cặp xxxx có chiều dài là 4 nên ta sử dụng {4}.

```
var pattern = /^[0-9]{4}-[0-9]{4}-[0-9]{4}-[0-9]{4}$/;  
if (pattern.test("1234-1232-1321-2312")) {  
    document.write('Mã thẻ hợp lệ');  
}  
else {  
    document.write('Mã thẻ không hợp lệ');  
}
```

Biểu thức chính quy

1. Khớp một Tên người dùng

Matching a username:



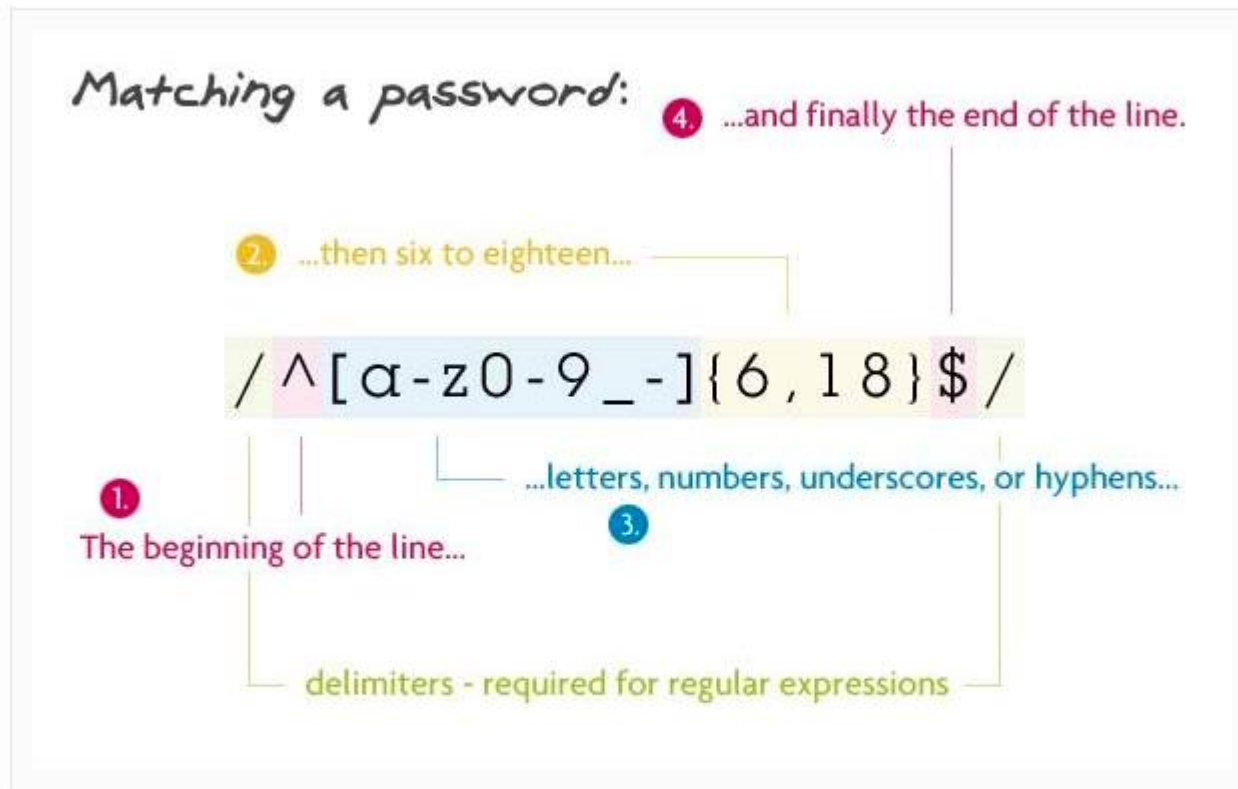
`/^[a-z0-9_-]{3,16}$/`

Biểu thức chính quy

phân tích theo cú pháp tìm đầu chuỗi (^), tiếp theo là bất kỳ chữ cái thường (a-z), con số (0-9), một dấu gạch dưới, hoặc một dấu gạch nối. Tiếp theo, {3,16} đảm bảo có ít nhất 3 trong số các ký tự đó, nhưng không quá 16. Cuối cùng, chúng ta kết thúc chuỗi (\$) .

Biểu thức chính quy

2. Khớp một Mật khẩu



Khớp một mật khẩu rất giống với việc khớp một tên người dùng.

Sự khác biệt duy nhất là thay vì 3 đến 16 chữ cái, các con số, dấu gạch dưới hoặc dấu gạch ngang, thì chúng ta muốn 6 đến 18 trong số đó ({6,18}).

Mẫu:

```
1 /^[a-z0-9_-]{6,18}$/
```

Biểu thức chính quy

3. Khớp một Giá trị Hex

Matching a hex value:

The beginning of the line... 1.

...followed by zero or one... 2.

5. ...letters or numbers... 8.

7. ...exactly three...

`/^#?([a-f0-9]{6}|[a-f0-9]{3})$ /`

3. ...number sign(s)...

4. ...then exactly six...

6. ...Or...

9. ...and finally the end of the line.

delimiters - required for regular expressions

Mẫu:

```
1 | /^#?([a-f0-9]{6}|[a-f0-9]{3})$ /
```

Biểu thức chính quy

cú pháp tìm đầu chuỗi (^). Tiếp theo, một ký hiệu con số là tùy chọn bởi vì nó được theo sau một dấu chấm hỏi. Dấu chấm hỏi nói cho trình phân tích cú pháp rằng ký tự trước — trong trường hợp này một ký hiệu số — là tùy chọn, nhưng phải nắm bắt nó nếu nó ở đó. Tiếp theo, bên trong nhóm đầu tiên (nhóm dấu ngoặc đơn đầu tiên), chúng ta có thể có hai tình huống khác nhau. Đầu tiên là bất kỳ chữ cái cái thường giữa a và f hoặc một con số sáu lần. Dấu gạch đứng cho chúng ta biết rằng chúng ta cũng có thể có ba chữ cái viết thường giữa a và f hoặc con số. Cuối cùng, chúng ta kết thúc chuỗi (\$) .

Biểu thức chính quy