Groovy regular expression tutorial

==~ operator

The match operator.The operator return true or false depend on the pattern match result. Here is an example:

```
println "great" ==~ /great/
```

Program output : true.

First we should familiar with the basic syntax of regular expression:

x?: Matches one or zero x.

x*: Matches zero or multiple x.

x+: Matches one or multiple x.

x{n}: Matches x repeat n times. a{4} represent aaaa.

x|y: Matches x or y.

[xyz]: Equals x|y|z

[x-z]: Matches anyone between x and y

^a: Means any character that is not a.

Match number

Suppose our string like

```
def s = '8902.33'
```

It takes 3 steps to match the string, match the integer part, match the point, match the number after point. Digit match pattern is /[0-9]+/, the point and the digit after point should be matched together, the pattern is /\.[0-9]+/. Here the \. is the escape of point because the point itself is a wildcard. The point is optional, so the pattern should be /(\.[0-9]+)?/. The final result is

```
def exp = /[0-9]+(\.[0-9]+)?/
println s ==~ exp
```

Output : true.

Match email address

```
def email = 'groovy@gmail.com'

def exp =  /[a-zA-Z][^@\.]+@[^@\.]+\.[^@\.]+/

println email ==~ exp
```

The expression match the first character that is alphabet and then any character that not @ and "." repeat one or multiple times , then the "@", then match the domain which separated by a ".".

If you want the match specific email service provider like google and yahoo:

```
def exp =  /[a-zA-Z][^@\.]+@((gmail.com)|(yahoo.com))/
```

Match phone number

```
def phone = '12234256785'
def exp = /12[0-9]{9}/

println phone ==~ exp
```

=~ operator

Also called find operator, it returns a java.util.regex.Matcher. A Matcher will search the string with a regular expression and return match result. The result include the whole match and the submatch.

```
    def link = '<a href="http://google.com/">Google</a>'
    def matcher = link =~ /<[^>]*?href="([\s\S]*?)"[\s\S]*/
    println matcher
    println matcher[0][0]
    println matcher[0][1]
```

The pattern match a html link, the submatch match the href attribute of the link. The output:

```
java.util.regex.Matcher[pattern=<[^>]*?href="([\s\S]*?)"[\s\S]* region=0,39 lastmatch=]
<a href="http://google.com/">Google</a>
http://google.com/
```

~string operator

The pattern operator returns a java.util.regex.Pattern from a pattern string.

```
    def pattern = ~/<[^>]*?href="([\s\S]*?)"[\s\S]*/
    def m = pattern.matcher(link)
    println m[0][0]
    println m[0][1]
```