

Korean Regular Expressions

Groovy `Pattern` . Groovy `Matcher` `=~` `==~` .

JSR-03 , `Matcher` `matcher[index]` `List` .

```
import java.util.regex.Matcher
import java.util.regex.Pattern

assert "cheesecheese" =~ "cheese"
assert "cheesecheese" =~ /cheese/
assert "cheese" == /cheese/ /* String */

// Pattern
def pattern = ~/foo/
assert pattern instanceof Pattern
assert pattern.matcher("foo").matches()

// Matcher
def matcher = "cheesecheese" =~ /cheese/
assert matcher instanceof Matcher
answer = matcher.replaceAll("edam")

//
def cheese = ("cheesecheese" =~ /cheese/).replaceFirst("nice")
assert cheese == "nicecheese"

// .
// match . API
// =~ Matcher .
// Matcher .
// matcher[0][1] 0 1 :
def m = "foobarfoo" =~ /o(b.*r)f/
assert m[0][1] == "bar"

// .
matcher = "\$abc." =~ "\\\$(.*)\\"
matcher.matches(); // [: true ? jsr-04 false .]
assert matcher.group(1) == "abc" // 0 1
// assert matcher[1] == "abc" // jsr-03-release true
assert matcher[0] == ["\$abc.", "abc"] // jsr-03-release true
assert matcher[0][1] == "abc" // jsr-03-release true
```

"/" .

```
def matcher = "\$abc." =~ /\$(.*)\./
assert "\\\$(.*)\\" == /\$(.*)\./
matcher.matches(); //
assert matcher.group(1) == "abc" // 0 1
// assert matcher[1] == "abc" // jsr-03-release true
assert matcher[0] == ["\$abc.", "abc"] // jsr-03-release true
assert matcher[0][1] == "abc" // jsr-03-release true
```

`Matcher` **find** `boolean` `if` `while` `=~` Perl `=~` . `==~` .

API .