

Regular Expression & Pattern Class

정규 표현식 작성 방법

기호	설명
[]	[abc] a, b, c 중 하나의 문자
	[^abc] a, b, c 이외의 하나의 문자
	[a-zA-Z] a~z, A~z 중 하나의 문자
\d	한 개의 숫자, [0-9]와 동일
\s	공백
\w	한 개의 알파벳 또는 한 개의 숫자
?	없음 또는 한 개
*	없음 또는 한 개 이상
+	한 개 이상
{n}	정확히 n개
{n,}	최소한 n개
{n, m}	n~m 개 까지
()	그룹핑

<http://palpit.tistory.com>

다음은 02-123-4567 또는 010-1234-4321 과 같은 전화번호를 위한 정규 표현식입니다.

기호	설명
(02 010)	02 또는 010
-	- 포함
\d{3,4}	3 또는 4자리 숫자
\d{4}	4 자리 숫자

다음은 zhfdi4@naver.com과 같은 이메일을 위한 정규 표현식입니다.

```
1 \w+@\w+\.\w+(\.\w+)?
2
```

CS

기호	설명
\w+	한 개 이상의 알파벳 또는 숫자
@	@
\.	.
(\.\w+)?	\.\w+이 없거나 한 번 더 올 수 있음

Pattern 클래스

정규 표현식으로 문자열을 검증하는 방법을 알아보도록 합시다. 문자열을 정규 표현식으로 검증하는 기능은 `java.util.regex.Pattern` 클래스의 정적 메소드인 `matches()` 메소드가 제공합니다.

```
1 boolean result = Pattern.matches("regular Expression", "String");
2
```

Colored by Color Scripter CS

첫 번째 파라미터는 정규 표현식이고, 두 번째 파라미터는 검증할 문자열입니다.

다음 예제는 전화번호와 이메일을 검증하는 코드를 보여줍니다.

* PatternExam.java

```
1 package api;
2
3
4 import java.util.regex.Pattern;
5
6 public class PatternExam {
7     public static void main(String[] args) {
8
9         String regExp = "(02|010)-\\d{3,4}-\\d{4}"; // Regular Expression
10        String data = "010-1234-4321";
```

CS

```

11
12     boolean result = Pattern.matches(regExp, data);
13
14     if (result)
15         System.out.println("정규식 일치");
16     else
17         System.out.println("정규식 불일치");
18
19     regExp = "\\w+@\\w+\\.\\w+(\\.\\w+)?";
20     data = "ios10@apple.co.m.";
21
22     result = Pattern.matches(regExp, data);
23
24     if (result)
25         System.out.println("정규식 일치");
26     else
27         System.out.println("정규식 불일치");
28
29 }
30 }
31
32
33

```

Colored by Color Scripter

