

정규표현식



정규 표현식이란?

문자열이 정해진 패턴으로 구성되어 있는지 검증해야 할 때, 사용합니다.
ex) 이메일, 전화번호

메타문자	기능	설명
수직선()	또는(or)	항목들 중 하나를 선택하기 위해 구분합니다. 예를 들면 "gray grey"는 "gray" 또는 "grey"와 일치합니다.
괄호()	그룹 묶기	연산자의 범위와 우선순위를 지정할 수 있습니다. 예를 들면 "gray grey"와 "gr(ale)y"는 "gray" 또는 "grey" 집합 둘 다 의미하는 같은 패턴입니다.
물음표(?)	0회 또는 1회	0번 또는 1번까지의 발생을 의미합니다. 예를 들면 "colou?r"는 "color"와 "colour" 둘 다 일치 시킵니다.
별표(*)	0회 이상	0번 이상 발생을 의미합니다. 예를 들면 "ab*c"는 "ac", "abc", "abbc", "abbbc" 등을 일치 시킵니다.
덧셈기호(+)	1회 이상	1번 이상의 발생을 의미합니다. 예를 들면 "ab+c"는 "abc", "abbc", "abbbc" 등을 일치 시킵니다. 그러나 "ac"는 일치시키지 않습니다.
{n}	n회	정확히 n번만큼 일치시킵니다.
{m, }	m회 이상	m번 이상만큼 일치시킵니다.
{m, n}	m회 이상 n회 이하	적어도 m번만큼 일치시킵니다. 그러나 n번을 초과하여 일치시키지는 않습니다. "a{1,3}b"는 "ab", "aab", "aaab"를 포함하지만, "b"나 "aaaab"는 포함하지 않습니다.

소제목

메타문자	기능	설명
.	문자	문자 한 개와 일치시킵니다.
[]	문자들	여러 문자들 중 하나의 문자와 일치시킵니다.
[^]	부정	해당 문자들을 포함하지 않는 문자들을 찾습니다.
^	처음	처음 시작하는 부분을 의미합니다.
\$	끝	끝나는 부분을 의미합니다.
()	하위 식	하위 식을 정의합니다.
\w	"_"와 영숫자	"_"를 포함한 영문자와 숫자를 일치시킵니다.
\W	\w 반대	"_"와 영문자 그리고 숫자를 제외한 다른 문자열들을 일치 시킵니다.
\s	공백	공백 문자와 일치시킵니다.
\S	공백 제외	공백을 제외한 어떤 것이든 일치시킵니다.
\d	숫자	숫자와 일치시킵니다.
\D	숫자 제외	숫자가 아닌 항목을 일치시킵니다.

소제목

리턴타입	메서드와 설명
static Pattern	compile(String regex) 주어진 정규 표현식을 이용하여 패턴으로 컴파일 합니다.
static Pattern	compile(String regex, int flags) 주어진 플래그 값과 정규 표현식을 이용하여 패턴으로 컴파일 합니다. 플래그 값은 CASE_INSENSITIVE, MULTILINE, DOTALL, UNICODE_CASE, CANON_EQ, UNIX_LINES, LITERAL, UNICODE_CHARACTER_CLASS 그리고 COMMENTS 등을 가질 수 있으며 비트연산자를 이용하여 여러 개 지정할 수 있습니다
리턴타입	메서드와 설명
Matcher	matcher(CharSequence input) 이 패턴과 지정한 입력을 매치하는 정규 표현 엔진을 만듭니다.
리턴타입	메서드와 설명
boolean	find() 문자열에서 다음 매치하는 문자열을 찾습니다.
boolean	find(int index) 문자열의 index위치 이후부터 다음 매치하는 문자열을 찾습니다.
String	group() 이전 match에서 찾은 문자열을 반환합니다.

Pattern클래스, Matcher클래스

Pattern클래스

-java.util패키지의 Pattern클래스의 matcher(), matchers() 메서드를 정규표현식 검증에 이용합니다

static메서드

compile(패턴) – 정규표현엔진(정규식)을 만들어 냅니다

matcher(검증 데이터) : 데이터를 비교해서 Matcher클래스로 반환

matchers(패턴, 검증 데이터) : 패턴으로 비교하여 boolean형으로 반환

```
String info = "30세/서울시 강남구/02-234-5678/010-234-5678";  
String pattern = "\\d{2,3}-\\d{3,4}-\\d{3,4}";  
  
Pattern p = Pattern.compile(pattern);  
Matcher m = p.matcher(info);
```

예시

Matcher클래스

find() : 해당 정규표현을 찾아냄

start() : 해당 정규표현의 시작 인덱스 반환

group() : 해당 정규표현의 값 반환

end() : 해당 정규표현의 끝 인덱스 반환



참고

자주 사용되는 패턴은 꼭 알아 두자!

`\\d{3}` 연속된 숫자 3개를 찾음

`\\w` 문자를 찾음

`\\w+` 문자 여러개

`[0-9]` 0-9 사이 문자를 찾음

`[a-z]` a-z 사이 문자를 찾음

`\\. \\[\\?` . [? 등 특정 문자를 찾음



Chapter 22

수고하셨습니다