

프로그래밍 언어 활용 강의안

(정규표현식)

정규식

문자열이 정해져 있는 형식(정규 표현식 : Regular Expression)으로 구성되어 있는지 검증해야 할 때 사용
사용자가 제대로 입력했는지 검증해야 할 때 정규 표현식 으로 비교

Ex) 이메일, 전화번호,ID,PW

문자 또는 숫자 기호와 반복 기호가 결합된 문자열

기호	설명		
[]	한 개의 문자	[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, [a-zA-Z_0-9]와 동일		
?	없음 또는 한 개		
*	없음 또는 한 개 이상		
+	한 개 이상		
{n}	정확히 n 개		
{n,}	최소한 n 개		
{n, m}	n 개에서부터 m 개까지		
()	그룹핑		

정규식

사용 예제

02-123-1234 또는 010-1234-5678 등의 전화번호를 위한 정규 표현식
(02|010)-\d{3,4}-\d{4}

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

abcd@naver.com 등의 이메일 표현 정규식
\w+ @\w+\w.\w+(\w.\w)?

\w+	한 개이상의 알파벳 또는 숫자
@	@문자 포함
\w+	한 개 이상의 알파벳 또는 숫자
\w.	. 문자 포함
\w+	한 개 이상의 알파벳 또는 숫자
(\w.\w+)?	\w.\w+ 이 없거나 한번 더 올 수 있음

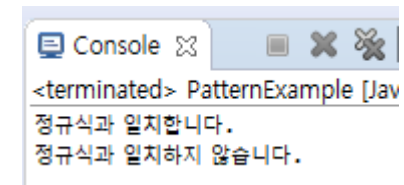
정규식

Pattern class

정규 표현식으로 문자열을 할 수 있는 함수 포함

Boolean result = Pattern.matches("정규식","검증할 문자열");

```
1
2 import java.util.regex.Pattern;
3
4 public class PatternExample {
5     public static void main(String[] args) {
6         String regExp = "(02|010)-\\d{3,4}-\\d{4}";
7         String data = "010-123-4567";
8         boolean result = Pattern.matches(regExp, data);
9         if(result) {
10             System.out.println("정규식과 일치합니다.");
11         } else {
12             System.out.println("정규식과 일치하지 않습니다.");
13         }
14
15         regExp = "\\w+@\\w+\\.\\w+(\\.\\w+)?";
16         data = "angel@navercom";
17         result = Pattern.matches(regExp, data);
18         if(result) {
19             System.out.println("정규식과 일치합니다.");
20         } else {
21             System.out.println("정규식과 일치하지 않습니다.");
22         }
23     }
24 }
25
```



Console

<terminated> PatternExample [Java]

정규식과 일치합니다.
정규식과 일치하지 않습니다.

정규식

정규식 테스트 사이트

<https://regexr.com/>

정규식 추천 사이트

<http://txt2re.com/index.php3>

JAVA 기반 정규식 (Doc)

<http://www.regexplanet.com/advanced/java/index.html>