

ApiPattern

정규표현식(Regular Expression)

- 특정 규칙을 가진 문자열의 집합을 표현하기 위해 쓰는 표현식
- 회원가입 시 아이디, 비밀번호가 요구하는 조건을 충족하는지 검사할때 쓰임

유효성 검사

^: 문자열의 시작

\$: 문자열의 끝

[]: 하나의 문자

+: 한 개 이상

*: 0 개 이상

1. 숫자만 존재하는지 검증

```
String regex = "[0-9]+$";
```

2. 알파벳만 존재하는지 검증

```
regex = "[a-zA-Z ]+$";
```

3. 한글만 존재하는지 검증

```
regex = "[가-힣|ㄱ-ㅎ|ㅏ-ㅣ]+$";
```

4. 아이디 패턴(영어+숫자, 길이는 7~12글자 사이)

```
regex = "[a-zA-Z0-9]{7}$";           // 7글자만 허용
regex = "[a-zA-Z0-9]{7,}$";         // 7글자 이상만 허용
regex = "[a-zA-Z0-9]{0,12}$";       // 12글자 이하
regex = "[a-zA-Z0-9]{7,12}$";       // 7~12 글자
```

```
regex = "^[\\w]{7,12}$";    //    \\w : 한 개의 알파벳
result = example.matches(regex);
```

5. 비밀번호 패턴 -1 (영어+숫자, 특수문자 포함)

```
regex = "^[\\w|!@#$%^&*()_+|]+$";
```

6. 비밀번호 패턴 -2

(영어 대+소문자, 숫자, 특수문자 각 1개 이상 포함 및 8~16사이)

```
regex = "^(?=.*[a-z])(?=.*[A-Z])(?=.*[0-9])(?=.*[!@#$%^&*()_+|])"
```

7. 이메일 형식 체크

```
regex = "^[\\w]+@[a-z]+[.][a-z]+$";
```

8. PIN 비밀번호 패턴 (동일 숫자 연달아 3회 이상 중복 금지)

```
regex = "^.*([0-9])\\1\\1.*$";
```