

## 정규식 패턴

- **^ (caret)**

라인의 처음이나 문자열의 처음을 표시

예 : ^aa (문자열의 처음에 aa 를 포함하면 참, 그렇지 않으면 거짓)

- **\$ (dollar)**

라인의 끝이나 문자열의 끝을 표시

예 : aaa\$ (문자열의 끝에 aaa 를 포함하면 참, 그렇지 않으면 거짓)

- **.** (period)

임의의 한 문자를 표시

예 : ^a.c (문자열의 처음에 abc, adc, aZc 등은 참, aa 는 거짓)

예 : a.b\$ (문자열의 끝에 aaab, abbb, azzb 등을 포함하면 참)

- **[] (bracket)**

문자의 집합이나 범위를 나타냄, 두 문자 사이의 범위는 "-" 사용.

[]내에서 "^"이 선행되면 not 을 나타냄

예 : [abc] (a, b, c 중 어떤 문자, "[a-c]."과 동일)

예 : [Yy] (Y 또는 y)

예 : [A-Za-z0-9] (모든 알파벳과 숫자)

예 : [-A-Z]. ("-"(hyphen)과 모든 대문자)

예 : [^a-z] (소문자 이외의 문자)

예 : [^0-9] (숫자 이외의 문자)

- **{ } (brace)**

{ } 내의 숫자는 직전의 선행 문자가 나타나는 횟수, 범위를 나타냄

예 : a{3} ('a'의 3 번만 반복인 aaa 만 해당됨)

예 : a{3,} ('a'가 3 번 이상 반복인 aaa, aaaa, ... 등을 나타냄)

예 : a{3,5} (aaa, aaaa, aaaaa 만 해당됨)

예 : ab{2,3} (abb 와 abbb 만 해당됨)

예 : [0-9]{2} (두 자리 숫자)

예 : doc[7-9]{2} (doc77, doc87, doc97 등이 해당)

예 : [^Zz]{3} (Z 와 z 를 포함하지 않는 5 개의 문자열, abc, ttt 등)

예 : .{3,4}er ('er'앞에 세 개 또는 네 개의 문자를 포함하는 문자열이므로 Peter, mother 등이 해당)

- **\* (asterisk)**

"\*" 직전의 선행 문자가 0 번 또는 여러 번 나타나는 문자열

예 : ab\*c ('b'를 0 번 또는 여러 번 포함하므로 ac, abc, abbbc 등)

예 : \* (선행 문자가 없는 경우이므로 임의의 문자열 및 공백 문자열)

예 : .\* (선행 문자가 "."이므로 하나이상의 문자를 포함하는 문자열)

예 : ab\* ('b'를 0 번 또는 여러 번 포함하므로 a, abcc, abb 등)

예 : a\* ('a'를 0 번 또는 여러 번 포함하므로 k, kd, a, aa, abb 등)

예 : doc[7-9]\* (doc7, doc777, doc778989, doc 등이 해당)

예 : [A-Z]\* (대문자로만 이루어진 문자열)

예 : like.\* (직전의 선행 문자가 '.'이므로 like 에 0 또는 하나 이상의 문자가 추가된 문자열이 됨, like, likely, liker, likelihood 등)

- **+ (Plus Sign)**

"+" 직전의 선행 문자가 1 번 이상 나타나는 문자열

예 : ab+c ('b'를 1 번 또는 여러 번 포함하므로 abc, abcd, abbc 등)

예 : ab+ ('b'를 1 번 또는 여러 번 포함하므로 ab, abcc, abb 등)

예 : [A-Z]+ (대문자로만 이루어진 문자열)

예 : like.+ (직전의 선행 문자가 '.'이므로 like 에 하나 이상의 문자가 추가된 문자열이 됨, likely, liker, likelihood 등, 그러나 like 는 해당 안됨)

- **? (question mark)**

"?" 직전의 선행 문자가 0 번 또는 1 번 나타나는 문자열

예 : ab?c ('b'를 0 번 또는 1 번 포함하므로 abc, abcd 만 해당됨)

- **() (parenthesis)**

()는 정규식내에서 패턴을 그룹화 할 때 사용

- **| (bar)**

or 를 나타냄

예 : a|b|c (a, b, c 중 하나, 즉 [a-c]와 동일함)

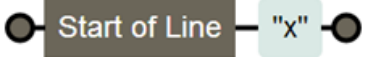
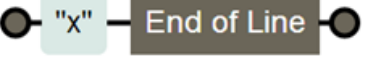
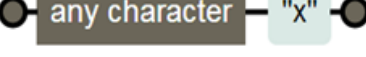
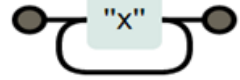
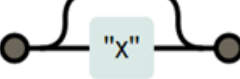


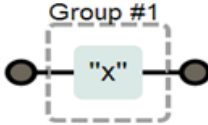
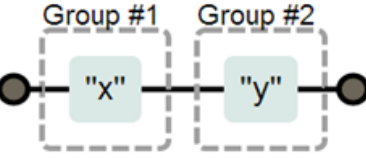
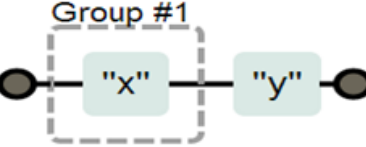
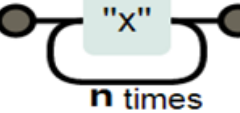
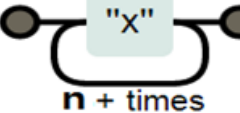
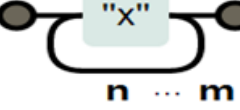
예 : yes|Yes (yes 나 Yes 중 하나, [yY]es 와 동일함)

예 : korea|japan|chinese (korea, japan, chinese 중 하나)

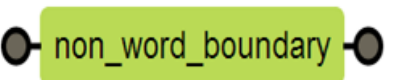

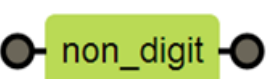
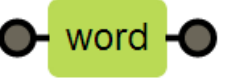
- **\ (backslash)**

위에서 사용된 특수 문자들을 정규식내에서 문자로 취급하고 싶을 때 '\ '를 선행시켜서 사용하면 됨

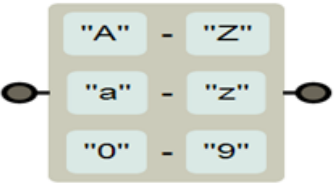
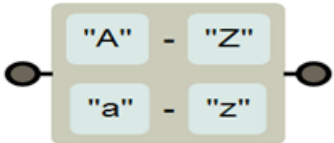



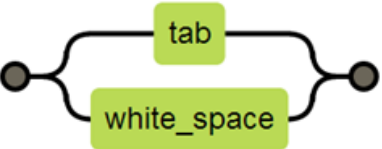
예 : filename\ .ext ("filename.ext"를 나타냄)    예 : [\?W[WWW]] ('?', '[', 'W', '] ' 중 하나)

정규표현식	표현	설명
$^x$		문자열이 x로 시작합니다.
$x\$$		문자열이 x로 끝납니다.
$.x$		임의의 한 문자를 표현합니다. (x가 마지막으로 끝납니다.)
$x^+$		x가 1번이상 반복합니다.
$x^?$		x가 존재하거나 존재하지 않습니다.
$x^*$		x가 0번이상 반복합니다.
$x y$		x 또는 y를 찾습니다. (or연산자를 의미합니다.)
$(x)$		( )안의 내용을 캡처하며, 그룹화 합니다.
$(x)(y)$		그룹화 할 때, 자동으로 앞에서부터 1번부터 그룹 번호를 부여해서 캡처합니다. 결과값에 그룹화한 Data가 배열 형식으로 그룹번호 순서대로 들어갑니다.
$(x)(?:y)$		캡처하지 않는 그룹을 생성할 경우 ?:를 사용합니다. 결과값 배열에 캡처하지 않는 그룹은 들어가지 않습니다.
$x\{n\}$		x를 n번 반복한 문자를 찾습니다.
$x\{n,\}$		x를 n번이상 반복한 문자를 찾습니다.
$x\{n,m\}$		x를 n번이상 m번이하 반복한 문자를 찾습니다.

문자클래스 '[' ]' : 그 내부에 해당하는 문자열의 범위 중 한 문자만 선택한다는 의미

정규표현식	표현	설명
[xy]		x,y중 하나를 찾습니다.
[^xy]		x,y를 제외하고 문자 하나를 찾습니다. (문자 클래스 내의 ^는 not을 의미합니다.)
[x-z]		x~z 사이의 문자중 하나를 찾습니다.
\w^		^(특수문자)를 식에 문자 자체로 포함합니다. (escape)
\wb		문자와 공백사이의 문자를 찾습니다.
\WB		문자와 공백사이가 아닌 값을 찾습니다.
\wd		숫자를 찾습니다.
\WD		숫자가 아닌 값을 찾습니다.
\ws		공백문자를 찾습니다.
\WS		공백이 아닌 문자를 찾습니다.
\wt		Tab 문자를 찾습니다.
\wv		Vertical Tab 문자를 찾습니다.
\ww		알파벳 + 숫자 + _ 를 찾습니다.
\WW		알파벳 + 숫자 + _ 을 제외한 모든 문자를 찾습니다.

POSIX 에서만 사용하는 문자클래스가 있는데, 대표적인 POSIX 문자클래스는 다음과 같으며 대괄호 '[' ] ' 가 붙어있는 모양 자체가 표현식이므로 실제로 문자클래스로 사용할 때에는 대괄호를 씌워서 사용해야만 정상적인 결과를 얻을 수 있다.

정규표현식	표현	설명
[[:alnum:]]		알파벳과 숫자를 찾습니다.
[[:alpha:]]		알파벳을 찾습니다.
[[:digit:]]		0~9사이를 찾습니다.
[[:lower:]]		알파벳 소문자를 찾습니다.
[[:upper:]]		알파벳 대문자를 찾습니다.
[[:blank:]]		탭과 공백문자를 찾습니다.