

○ 정규 표현식



- 패턴

표현식	의미
^x	문자열의 시작을 표현하며 x 문자로 시작됨을 의미한다.
x\$	문자열의 종료를 표현하며 x 문자로 종료됨을 의미한다.
.x	임의의 한 문자의 자리수를 표현하며 문자열이 x 로 끝난다는 것을 의미한다.
x+	반복을 표현하며 x 문자가 한번 이상 반복됨을 의미한다.
x?	존재여부를 표현하며 x 문자가 존재할 수도, 존재하지 않을 수도 있음을 의미한다.
x*	반복여부를 표현하며 x 문자가 0번 또는 그 이상 반복됨을 의미한다.
x y	or 를 표현하며 x 또는 y 문자가 존재함을 의미한다.
(x)	그룹을 표현하며 x 를 그룹으로 처리함을 의미한다.
(x)(y)	그룹들의 집합을 표현하며 앞에서 부터 순서대로 번호를 부여하여 관리하고 x, y 는 각 그룹의 데이터로 관리된다.
(x)(?y)	그룹들의 집합에 대한 예외를 표현하며 그룹 집합으로 관리되지 않음을 의미한다.
x{n}	반복을 표현하며 x 문자가 n번 반복됨을 의미한다.
x{n,}	반복을 표현하며 x 문자가 n번 이상 반복됨을 의미한다.
x{n,m}	반복을 표현하며 x 문자가 최소 n번 이상 최대 m 번 이하로 반복됨을 의미한다.

표현식	의미
[xy]	문자 선택을 표현하며 x 와 y 중에 하나를 의미한다.
[^xy]	not 을 표현하며 x 및 y 를 제외한 문자를 의미한다.
[x-z]	range를 표현하며 x ~ z 사이의 문자를 의미한다.
\w^	escape 를 표현하며 ^ 를 문자로 사용함을 의미한다.
\wb	word boundary를 표현하며 문자와 공백사이의 문자를 의미한다.
\WB	non word boundary를 표현하며 문자와 공백사이가 아닌 문자를 의미한다.
\wd	digit 를 표현하며 숫자를 의미한다.
\WD	non digit 를 표현하며 숫자가 아닌 것을 의미한다.
\ws	space 를 표현하며 공백 문자를 의미한다.
\WS	non space를 표현하며 공백 문자가 아닌 것을 의미한다.
\wt	tab 을 표현하며 탭 문자를 의미한다.
\vt	vertical tab을 표현하며 수직 탭(?) 문자를 의미한다.
\ww	word 를 표현하며 알파벳 + 숫자 + _ 중의 한 문자임을 의미한다.
\WW	non word를 표현하며 알파벳 + 숫자 + _ 가 아닌 문자를 의미한다.

- 플래그

Flag	Meaning	Description
i	Ignore Case	대소문자를 구별하지 않고 검색한다.
g	Global	문자열 내의 모든 패턴을 검색한다.
m	Multi Line	문자열의 행이 바뀌더라도 검색을 계속한다.

1.1 정규표현식

```
const reg = /.../
let strings = "ilikeyou";
document.write( strings.match( reg ) );
```

1.1 자주 사용하는 정규 표현식

숫자추출

```
const regexpr = /[0-9]+/g; // 0~9까지 1개이상
const regexpr = /[0-9,]+/g; // 0~9, ",", 1개이상
let regexpr = /[Wd,]+/g; // 숫자 , 1개이상
const regexpr = /^Wd{4}$/; // 4자리 숫자
```

특정단어 포함

```
const regexpr = /^http/; // http로 시작하는
const regexpr = /^[가-힣]+$/; // 한글만 포함
const regexpr = /^[a-zA-Z]+$/; // 알파벳만 포함
```

이메일 형식

```
/^[0-9a-zA-Z]([-_W.]?[0-9a-zA-Z])*@[0-9a-zA-Z]([-_W.]?[0-9a-zA-Z])*W.[a-zA-Z]{2,3}$/
```

^	문자열의 시작을 나타냅니다. (이메일이 문자열의 시작에서부터 일치해야 함을 의미)
[0-9a-zA-Z]	숫자(0-9), 소문자 알파벳(a-z), 대문자 알파벳(A-Z) 중 하나가 나와야 함을 의미합니다.
([-_W.]?[0-9a-zA-Z])*	[-_W.]? : 하이픈(-), 밑줄(_), 점(.) 중 하나가 0번 또는 1번 나타날 수 있음 -A, _1, .b 등이 여기에 해당 * 0번 또는 그 이상 반복
@	이메일 형식에서 '@' 기호가 반드시 포함되어야 함을 의미합니다.
[0-9a-zA-Z]	@ 다음에 숫자(0-9), 소문자 알파벳(a-z), 대문자 알파벳(A-Z) 중 하나가 나와야 함을 의미합니다.
([-_W.]?[0-9a-zA-Z])*	도메인 이름 부분에서도 하이픈(-), 밑줄(_), 점(.) 중 하나가 있을 수 있으며, 그 뒤에 숫자 또는 영문자가 오는 형태를 반복할 수 있음 (예: domain-name, domain.name 등 허용)
W.	도메인과 상위 도메인(.com, .net 등)을 구분하는 점(.)이 위치함을 의미합니다.
[a-zA-Z]{2,3}	[a-zA-Z] : 상위 도메인에 영문자(a-z, A-Z)만 허용 {2,3} : 상위 도메인이 2~3글자 (예: com, net, kr 등)
\$	문자열의 끝을 나타냅니다. (이메일이 문자열의 끝까지 일치해야 함을 의미)

핸드폰 번호

```
const regexpr = /^Wd{3}-Wd{3,4}-Wd{4}$/;
```

우편번호

```
const regexpr = /^Wd{5}$/;
```

url 형식

```
const regexpr = /^(https?|ftp):W/W/[^W$/.?#].[^W$]*$/;
```

공백검사

```
const regexpr = /^W$$/;
// Ws : 여러 가지 공백 문자 (스페이스, 탭 등)
// Wt : 탭, Wr : 줄바꿈, Wn :유닉스계열 줄바꿈, Wv:수직으로 이동하는 탭, Wf : 인쇄장치 페이지 넘김
```