

# JavaScript

---

---

---

---

# 정규표현식(Regular Expression)

- 역할

- ① 문자검색(search)
- ② 문자대체(replace)
- ③ 문자추출(extract)

- 생성

- 1) 생성자 함수 방식

```
const regexp1 = new RegExp("^abc");
```

- 2) 리터럴 방식

```
const regexp1 = /^abc/;
```

# 메소드

- 정규표현식에서 사용하는 메소드

메소드	문법	설명
exec	정규식.exec(문자열)	일치하는 하나의 정보(Array) 반환
test	정규식.test(문자열)	일치 여부(Boolean) 반환
match	문자열.match(정규식)	일치하는 문자열의 배열(Array) 반환
search	문자열.search(정규식)	일치하는 문자열의 인덱스(Number) 반환
replace	문자열.replace(정규식, 대체 문자)	일치하는 문자열을 대체하고 대체된 문자열(String) 반환
split	문자열.split(문자열)	일치하는 문자열을 분할하여 배열(Array)로 반환
toString	생성자_정규식.toString()	생성자 함수 방식의 정규식을 리터럴 방식의 문자열(String)로 반환

# 플래그 문자

- 기호

g	모든 문자와 여러 줄 일치(global)
i	영어 대소문자를 가리지 않고 비교(ignore case)
m	여러 줄 검사(multi line)
U	유니코드(Unicode)
Y	lastIndex 속성으로 지정된 인덱스에서만 1회 일치(sticky)

```
const regexp1 = new RegExp("^abc","gi");
```

# 정규식 패턴(표현식)

정규식 패턴	설명
^	줄(Line)의 시작에서 일치, /^abc/
\$	줄(Line)의 끝에서 일치, /xyz\$/
.	임의의 한 문자와 일치
a b	a 또는 b와 일치, 인덱스가 작은 것을 우선 반환
*	0회 이상 연속으로 반복되는 문자와 가능한 많이 일치, {0,}와 동일
*?	0회 이상 연속으로 반복되는 문자와 가능한 적게 일치(lazy), {0}와 동일
+	1회 이상 연속으로 반복되는 문자에 가능한 많이 일치, {1,}와 동일

# 정규식 패턴(표현식)

정규식 패턴	설명
+?	1회 이상 연속으로 반복되는 문자에 가능한 적게 일치 (lazy), {1}와 동일
?	없거나 1회 가능한 많이 일치
??	없거나 1회 가능한 적게 일치(lazy)
{3}	3(숫자)개 연속 일치
{3,}	3개 이상 연속 일치
{3,5}	3개 이상 5개 이하(3~5개) 연속 일치
{3,5}?	3개 이상 5개 이하(3~5개) 연속 중 가능한 적은 3개 연속 일치(lazy), {3}와 동일

# 정규식 패턴(표현식)

정규식 패턴	설명
()	캡처(Capture)할 그룹
(?<>)	캡처 그룹 이름 지정, /(?!<name>pattern)/ ES2018
₩1~9	정규식 내 캡처된 값 참조, /(abc)₩1/
(?:)	캡처(Capture)하지 않는 그룹
(?=)	앞쪽 일치(Lookahead), /ab(?!=c)/
(?!)	부정 앞쪽 일치(Negative Lookahead), /ab(?!c)/
(?<=)	뒤쪽 일치(Lookbehind), /(?!<=ab)c/ ES2018
(?<!)	부정 뒤쪽 일치(Negative Lookbehind), /(?!<!ab)c/ ES2018

# 정규식 패턴(표현식)

정규식 패턴	설명
[abc]	a 또는 b 또는 c와 일치, 점(.)이나 별표(*) 같은 특수 문자는 []안에서 특수 문자가 아님, /#\.[.]/
[a-z]	a부터 z 사이의 문자 구간에 일치(영어 소문자)
[A-Z]	A부터 Z 사이의 문자 구간에 일치(영어 대문자)
[0-9]	0부터 9 사이의 문자 구간에 일치(숫자)
[가-힣]	가부터 힣 사이의 문자 구간에 일치(한글)
[2-7]	2부터 7 사이의 문자 구간에 일치(2,3,4,5,6,7)
[b-f]	b부터 f 사이의 문자 구간에 일치(b,c,d,e,f)
[다-바]	다부터 바 사이의 문자 구간에 일치(다,라,마,바)



# 정규식 패턴(표현식)

정규식 패턴	설명
[^abc]	a 또는 b 또는 c가 아닌 나머지 문자에 일치(부정)
\w	이스케이프 문자, /\w.\w?\w/\w\$\w^/
\wb	63개 문자(영문 대소문자 52개 + 숫자 10개 + _(underscore))가 아닌 나머지 문자에 일치하는 경계(boundary)
\WB	63개 문자에 일치하는 경계
\wd	숫자(Digit)에 일치
\WD	숫자가 아닌 문자에 일치
\wp{}	유니코드 속성(Property) 집합에 맞는 문자에 일치, /\wp{Emoji}/u <b>ES2018</b>
\WP{}	유니코드 속성 집합에 맞지 않는 문자에 일치, /\wp{Uppercase}/u <b>ES2018</b>

# 정규식 패턴(표현식)

정규식 패턴	설명
<code>\s</code>	공백(Space, Tab 등)에 일치
<code>\S</code>	공백이 아닌 문자에 일치
<code>\w</code>	63개 문자(Word, 영문 대소문자 52개 + 숫자 10개 + _)에 일치
<code>\W</code>	63개 문자가 아닌 나머지 문자에 일치
<code>\x</code>	16진수 문자에 일치, <code>/\x61/</code> 는 a에 일치
<code>\0</code>	8진수 문자에 일치, <code>/\141/</code> 은 a에 일치
<code>\u</code>	유니코드(Unicode) 문자에 일치, <code>/\u0061/</code> 는 a에 일치
<code>\c</code>	제어(Control) 문자에 일치

# 정규식 패턴(표현식)

정규식 패턴	설명
\f	폼 피드(FF, U+000C) 문자에 일치
\n	줄 바꿈(LF, U+000A) 문자에 일치
\r	캐리지 리턴(CR, U+000D) 문자에 일치
\t	탭 (U+0009) 문자에 일치
\$`	문자 대체(replace) 시 일치한 문자 이전 값 참조
\$'	문자 대체(replace) 시 일치한 문자 이후 값 참조
\$+	문자 대체(replace) 시 마지막으로 캡처된 값 참조
\$&	문자 대체(replace) 시 일치한 문자 결과 전체 참조
\$_	문자 대체(replace) 시 입력(input)된 문자 전체 참조
\$1~9	문자 대체(replace) 시 캡처(Capture)된 값 참조

# 폼 유효성 검증

예시)

- 사용자가 필수적인 필드를 채웠는가?
- 사용자가 유효한 길이의 텍스트를 입력하였는가?
- 사용자가 유효한 이메일 주소를 입력하였는가?
- 사용자가 유효한 날짜를 입력하였는가?
- 사용자가 숫자 필드에 텍스트를 입력하지 않았는가?