

# 정규식 (regular expression)

텍스트 에디터에서 문자열을 찾기 & 바꾸기 기능을 생각해 보자. (find & replace 기능)  
Windows의 메모장의 찾기 & 바꾸기 기능은, 찾을 문자열과 완전히 동일한 문자열만 찾을 수 있다.

예를 들어 찾을 문자열이 **the** 이면 정확히 **the** 만 찾아지고 **The** 는 찾아지지 않는다.

문자열을 찾을 때, 완전히 동일한 문자열만 찾는 것이 아니고, 어떤 패턴과 일치하는 문자열을 전부 다 찾을 수 있으면 편리할 것이다.

찾을 문자열의 패턴을 지정하는 표현식(expression)이 정규식(regular expression) 이다.

예를 들어서 **the** 뿐만 아니라 **The** 도 찾을 수 있기 위한 정규식은 **[tT]he** 이다.  
eclipse, visual studio 등 대부분의 개발 도구는 정규식 찾기 기능을 지원한다.

## 정규식 특수 문자와 일반 문자

정규식에서 특수 문자가 아닌 평범한 문자는, 정확히 일치하는 것만 찾는다.

예를 들어 정규식 **[tT]he** 에서 **he** 부분은 평범한 문자이다. 그래서 he 부분은 정확히 일치해야 한다.

정규식에서 특수 문자는 찾을 패턴을 지정한다.

## [ ] 특수 문자

괄호 안의 문자들 중에서 아무 것이나 한 문자와 일치하는지 비교한다

예1) **[tT]he**

**[tT]** 부분은 **t** 문자와도 일치하고 **T** 문자와도 일치한다.

**th** 부분은 특수 문자가 아니기 때문에 정확히 일치해야 한다.

그래서 **[tT]he** 정규식은 **the** 와도 일치하고 **The** 와도 일치한다.

예2) **9[123][0123]**

**9** 부분은 특수 문자가 아니기 때문에 정확히 일치해야 한다.

일치하는 문자열들: 910 911 912 913 920 921 922 923 930 931 932 933

일치하지 않는 문자열들: 900 901 902 903 914 915 916 833

예3) **[123][0-3]8**

**8** 부분은 특수 문자가 아니기 때문에 정확히 일치해야 한다.

**[0-3]** 부분은 **[0123]** 과 동일하다.

따라서 일치하는 문자열들도 **[123][0123]8** 정규식과 동일하다.

예4) **[1-9][0-9]a**

a 부분은 특수 문자가 아니기 때문에 정확히 일치해야 한다.

**[1-9][0-9]** 정규식은 **[123456789][0123456789]** 정규식과 동일하다.

일치하는 문자열들: 12a 23a 35a 78a 90a 23a 30a 98a 81a 74a 99a

일치하지 않는 문자열들: 00a 01a 02a 03a 04a 05a 06a 07a 08a 09a 99b

예5) **0x[0-9A-F][0-9A-F]**

**0x[0-9A-F][0-9A-F]** 정규식은 **0x[0123456789ABCDEF][0123456789ABCDEF]** 정규식과 동일하다.

**0x** 부분은 특수 문자가 아니기 때문에 정확히 일치해야 한다.

일치하는 문자열들: 0x2F 0x34 0xAB 0xEE

일치하지 않는 문자열들: x2F 0x3f

## . 특수 문자

아무 문자 한 개와 일치한다.

예1) **A.B**

일치하는 문자열: AaB AzB A3B A@B A#B

일치하지 않는 문자열: AB AacB AxyzB

예1) **A..B**

일치하는 문자열: AabB AxzB A3aB A4@B A!#B

일치하지 않는 문자열: AB AaB AxyzB

## [^] 특수 문자

괄호 안의 문자들을 제외한 다른 문자들 중에서 아무 것이나 한 문자와 일치하는지 비교한다

예1) **[^tT]he**

일치하는 문자열: ahe bhe che Ahe Bhe Che 3he #he @he

일치하지 않는 문자열: the The

예2) **[^0-5][A-F]**

**[^0-5][A-F]** 정규식은 **[^012345][ABCDEF]** 정규식과 동일하다.

일치하는 문자열: 6A 7B 8A 9F AC aB bF

일치하지 않는 문자열: 0A 1B 2C 5F af bH 6Z

## ? 특수 문자

? 특수 문자 바로 앞의 정규식과 일치하는 문자 0 ~ 1 개 있는지 비교한다.

예1) **AB?C**

A C 부분은 정확히 일치해야 한다.

B? 부분은 B 문자가 0 ~ 1 개만 있는지 비교한다.

일치하는 문자열: AC ABC

일치하지 않는 문자열: ABBC ADC BBC

예2) **A[d-f]?C**

A C 부분은 정확히 일치해야 한다.

[d-f]? 부분은 [d-f] 정규식과 일치하는 문자가 0 ~ 1 개만 있는지 비교한다.

일치하는 문자열: AC AdC AeC AfC

일치하지 않는 문자열: AddC AEC

## \* 특수 문자

\* 특수 문자 바로 앞의 정규식과 일치하는 문자 0 ~ 여러 개 있는지 비교한다.

예1) **AB\*C**

A C 부분은 정확히 일치해야 한다.

B 부분은 B 문자가 0 ~ 여러 개 있는지 비교한다.

일치하는 문자열: AC ABC ABBC ABBBC ABBBBBC

일치하지 않는 문자열: ACC ADC BBC

예2) **A[d-f]\*C**

A C 부분은 정확히 일치해야 한다.

[d-f]\* 부분은 [d-f] 정규식과 일치하는 문자가 0 ~ 여러 개 있는지 비교한다.

일치하는 문자열: AC AdC AfeC AdeeC

일치하지 않는 문자열: AdgC AbeC AEC

## + 특수 문자

+ 특수 문자 바로 앞의 정규식과 일치하는 문자 1 ~ 여러 개 있는지 비교한다.

예1) **AB+C**

A C 부분은 정확히 일치해야 한다.

B+ 부분은 B 문자가 1 ~ 여러 개 있는지 비교한다.

일치하는 문자열: ABC ABBC ABBBC ABBBBC

일치하지 않는 문자열: AC CC ADC BBC

예2) **A[d-f]+C**

A C 부분은 정확히 일치해야 한다.

[d-f]+ 부분은 [d-f] 정규식과 일치하는 문자가 1 ~ 여러 개 있는지 비교한다.

일치하는 문자열: AdC AeC AfC AdefC AeefC

일치하지 않는 문자열: AC AEC ADEC A3378C

## ^ 특수 문자

^ 특수 문자는 줄바꿈 문자로 구분되는 각 줄의 선두 부분에서 있는 문자열인지 검사한다.

예1) **^123**

줄의 선두 부분에 있는 123 문자열과 일치한다.

줄의 선두가 아니면 일치하지 않는다.

## \$ 특수 문자

\$ 특수 문자는 줄바꿈 문자로 구분되는 각 줄의 끝 부분에서 있는 문자열인지 검사한다.

예1) **abc\$**

줄의 끝 부분에 있는 abc 문자열과 일치한다.

줄의 끝 부분이 아니면 일치하지 않는다.

## {n} 특수 문자

{ } 괄호 안에는 숫자가 1 개 들어 있어야 한다.

{ } 괄호 바로 앞의 정규식과 일치하는 문자가 반복되는 수가, { } 괄호 안의 숫자와 일치해야 한다.

예1) **AB{3}C**

일치하는 문자열: ABBBC

예2) **A[0-3]{4}C**

일치하는 문자열: A0000C A0123C A0023C A1320C A3211C

## **{n,m} 특수 문자**

{ } 괄호 안에는 숫자가 2 개 들어 있어야 한다.

{ } 괄호 바로 앞의 정규식과 일치하는 문자가 반복되는 수가, n 보다 크거나 같고 m 보다 작거나 같아야 한다.

예1) **AB{2,3}C**

일치하는 문자열: ABBC ABBBC

예2) **A[0-3]{2,4}C**

일치하는 문자열: A00C A013C A0023C A3211C

## **( ) 특수 문자**

( ) 괄호 안의 정규식을 묶는 역할을 한다.

예1) **A(BC)?D**

일치하는 문자열: AD ABCD

예2) **A(BC){2,3}D**

일치하는 문자열: ABCBCD ABCBCBCD

예3) **A([a-c][0-3]){2}D**

일치하는 문자열 Aa0b2D Ac2b0D Ab3c2D

일치하지 않는 문자열 Aba20D

예4) **A(23)+D**

일치하는 문자열: A23D A2323D A232323D

## | 특수 문자

| 특수 문자 앞 뒤의 정규식 둘 중의 하나와 문자열이 일치해야 한다.

예1) **Ab|cD**

| 특수 문자 앞의 정규식은 b 이고 뒤의 정규식은 c 이다.

위 정규식은 A[bc]D 정규식과 동일하다.

일치하는 문자열: AbD AcD

예2) **A(bc)|(def)G**

| 특수 문자 앞의 정규식은 ac 이고 뒤의 정규식은 def 이다.

일치하는 문자열: AbcG AdefG

## \ 특수 문자

정규식 특수 문자를 보통의 문자처럼 비교할 때, 특수 문자 앞에 \ 문자를 붙인다.

예1) **Ab?D**

일치하는 문자열: Ab?D

예2) **A[0-3]\+D**

일치하는 문자열: A0+D A1+D A2+D A3+D

예3) **\([1-9][0-9]\*\)**

일치하는 문자열: (1) (2) (3) (10) (21) (34) (100) (235)

일치하지 않는 문자열: (0) (01) [10]

예4) **\[1-9\]**

일치하는 문자열: [1-9]

예5) **[\t]**

탭 문자, 공백 문자 중의 한 문자와 일치한다.

예6) `[\t ]+`

탭 문자나 공백 문자가 1 개 이상 반복된 부분과 일치한다.

공백 문자 사이 사이에 탭 문자가 있어도 일치하고

탭 문자만 여러 개 반복되어도 일치하고

탭 문자 한 개나 공백 문자 한 개와도 일치한다.

## 연습 문제

정규식과 일치하는 배열 요소를 맞춰보세요

```
const regex = /^[a-zA-Z_][a-zA-Z0-9_]*$/;

const testStrings = [
  "variable",
  "",
  "_myVariable",
  "Name123",
  "test_1",
  "data_value",
  "1invalid",
  "invalid#",
  "AnotherValid_123"
];
```

답

```
const regex = /^[0-9]{6}-[1-4]{1}[0-9]{5}$/;

const testStrings = [
  "123456-123456",
  "654321-2-123456",
  "000000-123456",
  "123456-3-654321",
];
```

```
"123456-456789",  
"789012-4-987654",  
"123456-4-000000",  
"111111-1-222222"  
];
```

답

```
const regex = /^010-[0-9]{3,4}-[0-9]{4}$/;  
  
const testStrings = [  
  "010-123-4567",  
  "010-1234-5678",  
  "010-12-3456",  
  "010-12345-6789",  
  "010-123-456",  
  "011-123-4567",  
  "010-123-45678",  
  "010-123-4567-89"  
];
```

답

```
const regex = /^\[([1-9][0-9]*)\]$/;  
  
const testStrings = [  
  "(1)",  
  "(23)",  
  "(456)",  
  "(12a)",  
  "(7890)",  
  "(9)",  
  "(0)",  
];
```



```
"12",  
];
```

답

```
const regex = /^0x[0-9A-F]{2}$/;  
  
// 테스트할 문자열 배열  
const testStrings = [  
  "0x1A",  
  "0xFF",  
  "0x00",  
  "0xAB",  
  "0x34",  
  "0x1G",  
  "1A",  
  "0x123"  
];
```

답

```
const regex = /^%[0-9]*[dsfx]$/;  
  
const testStrings = [  
  "%d",  
  "%3s",  
  "%42f",  
  "%0x",  
  "%999d",  
  "%abc",  
  "%123g",  
  "% %d",  
  "%123"  
];
```

답

```
const regex = /^[a-zA-Z]+@[a-zA-Z]+\com$/;  
  
const testStrings = [  
  "example@domain.com",  
  "User@Company.com",  
  "test123@service.com",  
  "HelloWorld@sample.com",  
  "alice@wonderland.com",  
  "invalid@domain",  
  "@missingusername.com",  
  "missingatsign.com"  
];
```

답