

# 정규 표현식

# Regular Expressions

2023 전산언어학 겨울학교

# 차례

---

1. 정규 표현식이란?
2. 개별 문자 검색
3. 문자 집합 검색
4. 메타 문자
5. 반복 검색
6. 하위표현식
7. 전방/후방 탐색
8. 정규 표현식 실습



# 1. 정규 표현식이란?

## ◎ 정규 표현식(Regular Expression/Regex)

---

→ 텍스트를 찾고 조작하기 위해 사용하는 문자열

아래 예시들 모두 정규 표현식에 해당한다.

- Mary
- .
- [A-Za-z0-9]

## ◎ 정규 표현식을 왜 사용하는가?

---

(1) 검색(find)

→ 원하는 텍스트를 찾기

(2) 치환(replace)

→ 특정 텍스트를 찾아서 다른 텍스트로 바꾸기

## ◎ 유의사항

---

정규 표현식은...

- 어떤 특수한 프로그램이나 앱(application)이 아니다.
- 여러 프로그래밍 언어를 통해 실행된다.

## ◎ 정규 표현식 관련 사이트


---

RegexOne

<https://regexone.com/>

Regex101

<https://regex101.com/>



## 2. 개별 문자 검색



→ Mary와 같은 단순한 텍스트도 정규 표현식이다.

예시

---

Her name is Mary.

정규 표현식

---

Mary

결과

---

Her name is Mary.

→ 정규 표현식을 사용하여 동일한 텍스트 여러 개를 검색할 수 있다.

예시

---

I don't know your name. What is your name?

정규 표현식

---

name

결과

---

I don't know your name. What is your name?

## . (dot)

---

→ 마침표(.)는 모든 문자와 일치한다.

예시

---

Hello, world!

정규 표현식

---

.

결과

---

Hello, world!

→ 마침표(.)는 모든 문자와 일치하므로 다음과 같은 검색이 가능하다.

예시

---

BAT, BET, BUT

정규 표현식

---

B.T

결과

---

BAT, BET, BUT



# 3. 문자 집합 검색

## [ ] (brackets)

---

- 대괄호 [ ]는 문자 집합을 정의한다.
- [ ] 안에 있는 문자는 모두 집합의 원소가 된다.

ex) [A-Z] → A부터 Z까지 모든 대문자와 일치

ex) [a-z] → a부터 z까지 모든 소문자와 일치

ex) [0-9] → 0부터 9까지 모든 숫자와 일치

ex) [A-Za-z0-9] → 모든 대문자, 소문자, 숫자와 일치

→ A, B, C를 모두 검색하고 싶다면 집합 [ABC]를 구성하면 된다.

예시

---

Student A, Student B, Student C

정규 표현식

---

Student [ABC]

결과

---

Student A, Student B, Student C

→ 대문자와 소문자를 모두 검색하고 싶다면 다음과 같이 할 수 있다.

예시

---

NAME or name

정규 표현식

---

[Nn][Aa][Mm][Ee]

결과

---

NAME or name



→ [ ]를 사용하면 일일이 나열하는 것보다 더 간단하게 검색이 가능하다.

예시

---

ABCDEFGHIJKLMNOPQRSTUVWXYZ

정규 표현식

---

[A-Z]

결과

---

ABCDEFGHIJKLMNOPQRSTUVWXYZ

## ^(carat)

---

→ 캐럿(^) 문자는 검색 시에 제외하고 싶은 문자를 정할 때 사용한다.

※ 단, ^이 [ ] 내에서 사용되지 않을 때에는 문자열의 시작과 일치한다.

## 예시

---

ABCabc0123456789

## 정규 표현식

---

[^0-9]

## 결과

---

ABCabc0123456789

## 4. 메타 문자

## 메타 문자(metacharacters)

---

- 문자 그대로 사용되지 않고 특별한 용도로 사용되는 문자
- 다음의 문자들은 기초적인 메타 문자들이다.

.	모든 문자와 일치
[ ]	문자 집합을 구성하는 원소와 일치
[^]	문자 집합을 구성하는 원소를 제외하고 일치
-	범위 정의([ ]와 함께 사용)
\	뒤에 오는 문자를 이스케이프 (문자들이 문자 그대로 해석되도록 하는 기능)

## 이스케이프(escape)

---

→ 역슬래시(\)를 사용하여 뒤에 오는 문자를 문자 그대로 사용하는 것

Q: 앞서 마침표(.)는 모든 문자와 일치한다고 했다. 그럼 마침표를 그 자체로 찾아내려면 어떻게 해야 할까?

A: 해결책은 바로 마침표(.)를 이스케이프하는 것이다.

. → \.

※ 자판에 역슬래시(\)가 없으면 원화 기호(₩)를 사용하면 된다.

→ 마침표(.)를 찾으려면 마침표를 이스케이프하면 된다.

예시

---

I don't know your name. What is your name?

정규 표현식

---

\.

결과

---

I don't know your name. What is your name?

→ 역슬래시(\) 자체를 찾고 싶다면 역슬래시를 이스케이프하면 된다.

예시

---

\Student A\Student B\Student C\

정규 표현식

---

\\

결과

---

\Student A\Student B\Student C\

## 기타 메타 문자들

---

아래 메타 문자들은 유용하게 사용될 수 있다.

\w	영, 숫자 문자나 밑줄과 일치 [a-zA-Z0-9_]
\W	\w와 반대로 일치 [^a-zA-Z0-9_]
\d	모든 숫자와 일치
\D	\d와 반대로 일치
\b	단어 경계와 일치



# 5. 반복 검색

## + (plus)

---

→ 더하기(+)는 문자가 1개 이상일 때 일치한다.

예시

---

a, ab, abb, abbb

정규 표현식

---

ab+

결과

---

a, ab, abb, abbb

## \* (asterisk)

---

→ 별표(\*)는 문자가 없거나 1개 이상일 때(즉, 0개 이상일 때) 일치한다.

예시

---

a, ab, abb, abbb

정규 표현식

---

$ab^*$

결과

---

a, ab, abb, abbb

→ +와 \*를 문자 그 자체로써 찾으려면 이스케이프해야 한다.

예시

---

+ \*

정규 표현식

---

\+ \\*

결과

---

+ \*

$\{m\}$

---

→ 특정 요소가 정확히  $m$ 회 반복되는 경우에 일치한다.

예시

---

a, aa, aaa

정규 표현식

---

$a\{3\}$

결과

---

a, aa, **aaa**

$\{m,\}$

---

→ 특정 요소가 m회 이상 반복되는 경우에 일치한다.

예시

---

a, aa, aaa

정규 표현식

---

$a\{2,\}$

결과

---

a, aa, aaa

**$\{m,n\}$**

---

→ 특정 요소가 m회 이상 n회 이하 반복되는 경우에 일치한다.

예시

---

a, aa, aaa

정규 표현식


---

$a\{1,3\}$

결과

---

a, aa, aaa



## 6. 하위 표현식



## 하위 표현식(subexpression)

---

- 상위 표현식 내부의 특정 표현식을 하나의 패턴으로 취급하여 묶은 것
- 소괄호 ( )를 사용하여 나타낸다.

예시

---

ABCABCABC

정규 표현식

---

(ABC){3}

결과

---

ABCABCABC

- 하위 표현식을 사용하면 반복되는 패턴을 묶는 것이 가능하다.
- 즉, 아래와 같은 예시를 더 간단하게 풀 수 있다.

예시

---

12.143.33.150

정규 표현식

---

`\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}`

결과

---

12.143.33.150

→ 동일한 예시를 하위 표현식을 사용해 풀면 다음과 같다.

예시

---

12.143.33.150

정규 표현식


---

$(\backslash d\{1,3\}\backslash.){3}\backslash d\{1,3\}$

결과

---

12.143.33.150



# 7. 전방/후방 탐색

## 전방/후방 탐색

---

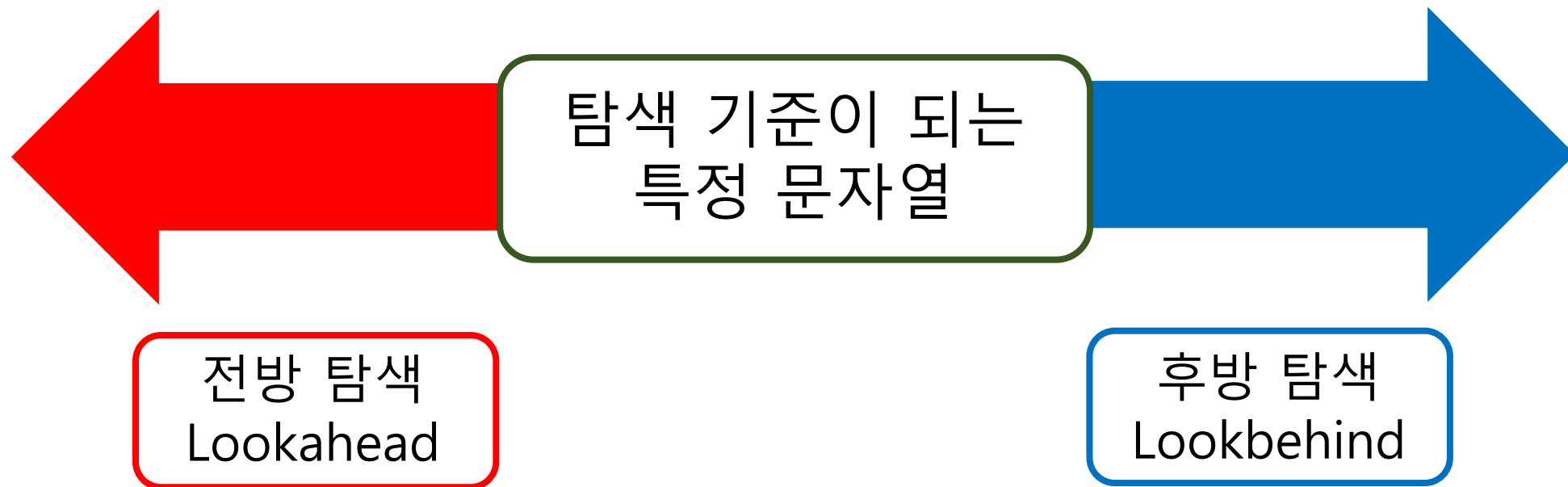
- 텍스트를 검색하다 보면 원하는 부분과 그렇지 않은 부분이 있다.
- 원하는 부분만을 일치시키기 위해 전방/후방 탐색이 필요하다.

(?=)	긍정형 전방 탐색 → 특정 문자열의 앞부분을 검색
(?<=)	긍정형 후방 탐색 → 특정 문자열의 뒷부분을 검색
(?!)	부정형 전방 탐색 → 앞에서 지정한 패턴과 일치하지 않는 텍스트를 검색
(?<!)	부정형 후방 탐색 → 뒤에서 지정한 패턴과 일치하지 않는 텍스트를 검색

## 전방/후방 탐색

---

→ 특정 문자열 기준으로 탐색 방향은 다음과 같다.



## 긍정형 전방 탐색(positive lookahead)

---

- 특정 문자열의 앞부분을 찾는 방법이다.
- 특정 문자열 자체는 검색 결과에 포함되지 않는다.

예시

---

ab, ac, ad, ae

정규 표현식

---

a(?=b)

결과

---

ab, ac, ad, ae

## 긍정형 후방 탐색(positive lookbehind)

---

- 특정 문자열의 뒷부분을 찾는 방법이다.
- 특정 문자열 자체는 검색 결과에 포함되지 않는다.

예시

---

\$1, \$10, \$100

정규 표현식

---

(?<=\\$)[0-9]+

결과

---

\$1, \$10, \$100



## 부정형 전방 탐색(negative lookahead)

---

→ 앞에서 지정한 패턴과 일치하지 않는 텍스트를 찾는 방법이다.

예시

---

ab, ac, ad, ae




정규 표현식

---

a(?!b)

결과

---

ab, ac, ad, ae

## 부정형 후방 탐색(negative lookbehind)

---

→ 뒤에서 지정한 패턴과 일치하지 않는 텍스트를 찾는 방법이다.

예시

---

I paid \$40 for 20 candy bars but I still have \$10.

정규 표현식


---

`\b(?<!\$)\d+\b`

결과

---

I paid \$40 for 20 candy bars but I still have \$10.



## 8. 정규 표현식 실습

# 유니코드 & 인코딩

→ 유니코드(Unicode)

: 전세계의 여러 문자를 컴퓨터에서 일관적으로 처리하기 위해 고안된 국제 표준 체계

→ 문자 인코딩(character encoding)

: 문자 혹은 기호를 컴퓨터가 처리할 수 있도록 신호로 변환하는 절차

# 한글 처리

→ 유니코드 체계에서 한글은 다음과 같이 정의된다.

<http://www.unicode.org/charts/PDF/UAC00.pdf>

ㄱ : U+3131

ㅎ : U+314E

ㅏ : U+314F

ㅣ : U+3163

가 : U+AC00

힉 : U+D7A3

/[가-힉]/ = /\uAC00-\uD7A3/

# UTF-8이란?

- UTF-8은 Universal Coded Character Set + Transformation Format – 8-bit의 줄임말이다.
- 자연어 처리에서는 인코딩을 UTF-8으로 통일한다.
- 따라서 본 실습 또한 UTF-8 인코딩을 기준으로 진행한다.

# 실습 자료 링크

신문 기사1

<https://n.news.naver.com/mnews/article/088/0000796450?sid=103>

신문 기사2

<https://n.news.naver.com/mnews/article/047/0002380963?sid=103>

**Thank you for listening!**