

# Regular Expressions

정규표현식은 특정 문자열을 가리키는 패턴이다. 즉 이 패턴으로 원하는 문자열을 찾는다.

- 켄 톰슨

## 기본 패턴 찾기

- 클래스; []
- 단축문자; \d: 숫자, \D: 숫자 이외의 문자
- 임의의 문자; .: 아무문자(행의 끝에 오는 개행 문자는 제외)
- 그룹참조; ()
- 수량자; {숫자}: 해당 갯수, {최소숫자,최대숫자}: 최소숫자부터 최대숫자 갯수만큼, ?: 하나 이하, +: 하나 이상, \*: 0 이상
- 상수

## 숫자

- \d, [0-9]

## 숫자가 아닌 문자

- \D, [^0-9]: 공백, 구두점, 인용부호, 하이픈, 슬래시, 대괄호 같은 문자도 찾는다.
- \w; 모든 영문자, 숫자, \_, 기타 스크립트 문자 = [a-zA-Z0-9\_]
- \W; [^a-zA-Z0-9\_]

| 단축 문자 <sup>1)</sup> |              |
|---------------------|--------------|
| \a                  | 벨 문자         |
| [\b]                | 백스페이스 문자     |
| \c x                | 제어 문자        |
| \d                  | 숫자           |
| \D                  | 숫자가 아닌 문자    |
| \d xxx              | 문자의 10진수 값   |
| \f                  | 폼 피드 문자      |
| \h                  | 수평 공백        |
| \H                  | 수평 공백이 아닌 문자 |
| \r                  | 캐리지 리턴       |
| \n                  | 개행 문자        |
| \0xxx               | 문자의 8진수 값    |
| \s                  | 공백 문자        |
| \S                  | 공백이 아닌 문자    |
| \t                  | 수평 탭 문자      |
| \v                  | 수직 탭 문자      |
| \V                  | 수직 탭이 아닌 문자  |

| 단축 문자 <sup>1)</sup> |                                              |
|---------------------|----------------------------------------------|
| <code>\w</code>     | 영문자, 숫자, <code>_</code> , 기타 스크립트 문자         |
| <code>\W</code>     | 영문자, 숫자, <code>_</code> , 기타 스크립트 문자를 제외한 문자 |
| <code>\0</code>     | 널 문자                                         |
| <code>\x xx</code>  | 문자의 16진수 값                                   |
| <code>\u xxx</code> | 문자의 유니코드 값                                   |

## 공백

- `\s`, `[ \t\n\r]`

## 임의의 문자

- `.` (`U+002E`); 개행 문자<sup>2)</sup>를 제외한 모든 문자
- `\b`; 단어의 경계

## 참조 그룹

- 찾고자 하는 내용을 ()로 감싼 후(참조) `$1`, `$2` 혹은 `\1`, `\2`와 같이 역참조

## 경계

- 위치 지정자 **assertion**
  - 행의 시작과 끝
  - 단어의 경계(두 종류)
  - 행의 시작과 끝
  - 문자열 상수를 나타내는 경계
- `^`; 문맥에 따라 행이나 문자열 또는 문서 전체의 시작
- `$`; 행이나 문자열의 끝
- `\b`; 단어의 경계
- `\B`
- `\<`; 단어의 시작
- `\>`; 단어의 끝
- `\A`; `^`의 기능처럼 해당 패턴이 행의 시작 위치에 나오는지 찾는다. PCRE(Perl Compatible Regular Expression)
- `\Z`, `\z`; 해당 패턴이 행의 끝에 나오는지

## 선택, 그룹, 역참조

## 문자 클래스

# 유니코드와 기타 문자

## 수량자

## 자주 쓰는 정규표현식

- 자음 모음을 포함한 모든 한글 검색;  
[ㄱ-ㅣ5가-히ㅎ]+
- 이메일 주소 확인;  
^[a-zA-Z][\w-.%]+@([\w-]{2,63}\.)+[a-zA-Z]{2,4}\$
  1. ^문장의 시작부터 \$끝까지 체크
  2. 대소문자 구분 없이 a-z,0-9,-,.,\_,의 1개 이상
  3. @
  4. 대소문자 구분 없이 a-z,0-9,-,.\_가 2글자 이상 63자 이하의 글자가 오며 점(.)은 1개 이상
  5. 대소문자 구분 없이 a-z의 2개 이상 4개 이하
- IP 주소체크;  
((25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)\.){3}(25[0-5]|2[0-4][0-9]|[01]?[0-9][0-9]?)
  1. 250~251 또는 200~249 또는 0~199 \. 정확히 3번
  2. 250~251 또는 200~249 또는 0~199 1번
- URL;  
^https?:\/\/([\w-]+.)+([\w-]?&%=]\*)?\$
  1. ^문장의 시작부터 \$끝까지 체크
  2. http로 시작하고 s는 있거나 없거나
  3. :// 체크
  4. zA-Z0-9 등 \w에 매칭되는 문자와 -은 1번 이상
  5. 점(.)
  6. 4,5번의 조합이 1번 이상
  7. / 이후 \w에 매칭되는 문자와 -./?&%= 조합의 0번 이상이 있거나 없거나
- 신용카드번호
  - 마스터카드(5로 시작);  
^5\d{3}-?\d{4}-?\d{4}-?\d{4}\$
  - 비자카드(4로 시작);  
^4\d{3}-?\d{4}-?\d{4}-?\d{4}\$
  - 국내전용(9로 시작);  
^9\d{3}-?\d{4}-?\d{4}-?\d{4}\$
  - 아메리칸익스프레스(3으로 시작, 두번째숫자는 4또는 7);  
^3[47]\d{2}-?\d{4}-?\d{4}-?\d{4}\$
    1. ^문장의 시작부터 \$끝까지 체크
    2. \d{4} 0-9까지의 숫자가 네자리
    3. -?-가 있거나 없거나
- HTML 주석;  
<!--{2,}.\*?--{2,}>
  1. <!로 시작
  2. -가 2개 이상
  3. .\*? 아무 문자가 0번 이상
  4. -가 2개 이상

## 5. >로 다음

# Tools

## On-line

- [RegEx Pal](#)
- [RegExr](#)

## Offline

- QED
- ed
- sed; s/pattern1/replacetext/ s(substitute)
- vi(vim)
- grep
- awk
- [TextMate](#) macOS
- [Notepad++](#) Windows
- [Oxygen XML Editor](#)

1)

이 외에 더 있음, 정규표현식 엔진에 따라 지원 여부 다양

2)

유닉스에서는 \n(U+0004), 윈도우에서는 \n과\r(U+000D)

From:

<http://wiki.theta5912.com/> - reth

Permanent link:

<http://wiki.theta5912.com/doku.php?id=public:computer:regexp&rev=1629710875>

Last update: **2021/08/23 18:27**

