

31장 RegExp (정규표현식)

31.1 정규 표현식이란 ?

- 자바스크립트 고유 문법 X
- 문자열 대상으로 패턴 매칭 기능 제공

31.2 정규 표현식의 생성

- 두 가지 생성방법
 - 정규 표현식 리터럴 — 대부분 이걸로 쓴다
 - RegExp 생성자 함수
- / regex / i
 - '/' 는 시작, 종료 기호
 - 'Regex' 는 패턴
 - 'i' 는 플래그

31.3 RegExp 메서드

1. RegExp.prototype.exec — 매칭 결과를 배열로 반환, 매칭 결과 없는 경우 null
g 플래그를 주어도 첫 번째 매칭 결과만 반환
 2. RegExp.prototype.test — 매칭 결과를 불리언 값으로 반환
 3. String.prototype.match — 매칭 결과를 배열로 반환
g 플래그가 주어지면 모든 매칭 결과 반환
- 이외에 string 메서드들이 있다. 나중에 살펴볼 것

31.4 플래그

- 정규 표현식의 검색 방식을 설정하기 위해 사용
- 총 6개로 주로 사용하는 건 3개이다.
 - i => (ignore case) 대소문자를 구분하지 않고 패턴을 검색
 - g => (global) 대상 문자열 내에서 패턴과 일치하는 모든 문자열을 전역 검색
 - m => (Multi line) 문자열의 행이 바뀌더라도 패턴 검색을 계속함.
- 동시에 여러개 설정 가능하다.
- 플래그 입력 안 할 시 대소문자 구별해서 패턴을 검색하고 검색 대상이 1개 이상 존재해도 첫 번째 매칭한 대상만 검색하고 종료

31.5 패턴

1. 문자열 검색

2. 임의의 문자열 검색

. 은 임의의 문자 한 개를 의미한다.

3. 반복 검색

- {m, n} 은 앞선 패턴이 최소 m번, 최대 n번 반복되는 문자열을 의미 — Ex) const regExp = /A{1,2}/g; — 주의!!! 콤마 뒤에 공백있으면 정상동작 x
- {n} 은 앞선 패턴이 n번 반복되는 문자열을 의미 즉, {n,n}과 같다.
- {n,} 앞선 패턴이 최소 n번 이상 반복되는 문자열
- +는 앞선 패턴이 최소 한 번 이상 반복되는 문자열을 의미 즉, +는 {1,}과 같다.
- ? 는 앞선 패턴이 최대 한 번(0번 포함) 이상 반복되는 문자열을 의미 즉, ?는 {0,1}과 같다.

4. OR 검색

| 는 or의 의미를 갖는다.

[] 내의 문자는 or로 동작한다.

범위를 지정하려면 [] 내에 -를 사용한다.

- [A-Za-z] — 모든 알파벳 검색
- [0-9] — 숫자 검색 — \d (= [0-9])는 숫자를 의미
- \w 는 알파벳, 숫자, 언더스코어 — \D 는 숫자가 아닌 문자를 의미
- \W 는 알파벳, 숫자, 언더스코어를 제외한 문자. — 공백, %& 등

5. NOT 검색

[...] 내의 ^ 는 not을 의미

6. 시작 위치로 검색

[...] 밖의 ^ 는 문자열의 시작을 의미

7. 마지막 위치로 검색

\$는 문자열의 마지막을 의미

31.6 자주 사용하는 정규표현식

1. 특정 단어로 시작하는지 검사 — ^ 을 사용
2. 특정 단어로 끝나는지 검사 — \$ 을 사용
3. 숫자로만 이루어진 문자열인지 검사 — /\d+\$/ .test('12345') // true
4. 하나 이상의 공백으로 시작하는지 검사 — /^[]+/.test(' Hil') // true — \s 는 여러 가지 공백 문자(스페이스, 탭 등)을 의미
5. 아이디로 사용 가능한지 검사 — /^[A-Za-z0-9]{4,10}\$/ .test('abc123') // true — 알파벳 소문자, 대문자 또는 숫자로 시작하고 끝나며 4~10자리인지 검사
6. 메일 주소 형식에 맞는지 검사 — /^[0-9a-zA-Z]([_]?[0-9a-zA-Z]) * @ [0-9A-Za-z]([_]?[0-9a-zA-Z]) * \.[a-zA-Z]{2,3}\$/
7. 핸드폰 번호 형식에 맞는지 검사 — /\d{3}-\d{3,4}-\d{4}\$/
8. 특수 문자 포함 여부 검사
 - /^[A-Za-z0-9]{1,}/.test('abc#123') // true
 - Replace 사용 특수문자 제거 — 'abc#123'.replace(/^[A-Za-z0-9]{1,}/g, '') // abc123