

# 소프트웨어융합기초1

## JavaScript

김경민



# 문자열

- 문자열.length : 문자열 길이
- 특정문자 추출
  - 문자열[인덱스] : 인덱스는 0부터 시작
  - 문자열.charAt[인덱스]
- 문자열 처음부터 끝까지 순회
  - for (let c of 문자열){}
- 문자인지 확인 및 숫자 변환
  - isNaN() : 유효한 숫자 값에 대해 False를 반환
  - parseInt() : 숫자로 변환
- 대소문자 변경
  - 문자열.toUpperCase(), 문자열.toLowerCase()
- 문자열에서 특정 문자열 찾기
  - 문자열.indexOf(" 찾을문자열 ") : 찾을 문자열의 시작위치 (-1이면 없음)
  - 문자열.indexOf(" 찾을문자열", 검색시작위치) : 검색시작위치에서 부터 찾을

- 문자열에서 특정 문자열 찾기
  - 문자열.includes(" 찾을문자열 ") : 있으면 true, 없으면 false
  - 문자열.includes(" 찾을문자열" 검색시작위치) : 검색시작위치에서 부터 찾아 : 있으면 true, 없으면 false
- 문자열 추출하기
  - slice
    - 문자열.slice(시작) : 시작위치에서 부터 끝까지 추출
    - 문자열.slice(시작, 종료) : 시작에서 종료전까지 추출
    - 시작과 종료 숫자가 음수이면 끝에서 부터 카운팅
- substring
  - slice와 아주 유사하지만 시작이 종료보다 커도 되지만 음수는 사용할 수 없음
- substr
  - 문자열.substr(시작, 길이) : 시작에서부터 길이개의 글자 반환

# 해결문제

## 문자열 다루기

단어를 입력하세요.

회문확인

숫자합계

취소

HTML



CSS



JavaScript

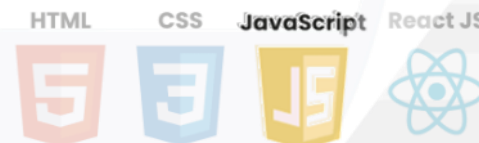


React JS



# 배열

- 순서가 있는 컬렉션을 저장
- 배열 선언 및 생성
  - `let arr = new Array();` 나 `let arr = [];`
- trailing 쉼표
  - 배열의 마지막 요소는 쉼표로 끝낼 수 있음
  - 모든 줄의 생김새가 유사해지기 때문에 새로운 속성을 추가할 때, 마지막 줄에 trailing comma가 있다면 그 줄을 수정 없이 그대로 복사해 쓸 수 있음
- 배열 크기
  - `배열명.length`
- 배열의 `length`는 수정이 가능
  - `arr.length = 0;`을 사용해 아주 간단하게 배열을 비울 수 있음



# 배열

- 배열 추가 삭제

- 배열의 끝에서 추가 삭제 : 배열명.push(), 배열명.pop()
- 배열의 앞에서 추가 삭제 : 배열명.unshift(), 배열명.shift()(shift는 제거요소 반환)

- 배열의 요소 접근

- 배열명[인덱스]

- 배열 요소 찾기

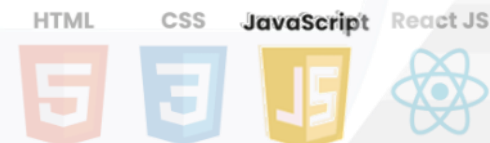
- arr.includes(찾을요소) : 배열에 찾을 요소가 있는지 확인

- 배열 순회

- 반복문 활용

- 배열 복사

- 전개연산자(...)를 활용



# 배열

## • 배열 정렬

- `arr.sort()` : 알파벳 순으로 정렬
- 숫자 정렬
  - `arr.sort((a, b) => a - b);` // 오름차순 정렬
  - `arr.sort((a, b) => b - a);` // 내림차순 정렬

## • 배열 요소 찾기

- `arr.indexOf()`

## • 배열 합치기

- `arr.join("")`

## • 구조분해할당(Destructuring assignment)

- 배열이나 객체의 요소 및 프로퍼티들을 분해해서 그 값들을 각각의 변수에 할당하는 자바스크립트의 표현식



# 배열

- 배열.filter(콜백함수)
  - 주어진 함수의 조건을 만족하는 요소를 모아 새로운 배열로 반환
- 배열.map(콜백함수)
  - 배열 내의 모든 요소 각각에 대하여 주어진 함수를 호출한 결과를 모아 새로운 배열을 반환
- 전개연산자(spread operation)
  - ... : 나열형 자료 추출하거나 연결

```
/* 실습 : 합집합, 교집합, 차집합 */
let setA = [1,2,3,4,5];
let setB = [4,5,6,7,8];

//합집합
let union = [...setA, ...setB];
console.log(union);

//교집합
let intersection = setA.filter((item) => setB.includes(item));
console.log(intersection);

//차집합
let difference = setA.filter((item) => !setB.includes(item));
console.log(difference);

//합집합 수정
union = [...difference, ...setB];
console.log(union);
```

JavaScript React JS



# 객체(object)

- 속성과 메서드를 가지는 데이터 타입
  - 중괄호({}) 안에 '키(key): 값(value)' 쌍으로 구성
  - 키는 문자형, 값은 모든 자료형이 허용
- 객체 선언 및 생성
  - `let obj = new Object();` , `let obj = {};`
- 객체 속성
  - `let obj = { name : "홍길동", age: 30, }`
- 객체 요소 접근
  - 점 표기법 : `obj.name`
  - 대괄호 표기법 : `obj["name"]`
- 객체 추가
  - `객체["새로운키"] = 값`
- 객체 삭제
  - `delete 객체.key`
- 객체 순회
  - 반복문 활용
- 객체 키와 값 접근
  - `Object.entries()`



# 실습과제

자바스크립트

w3schools HTML

w3schools JS

로또번호 생성기

번호생성하기

HTML



CSS



JavaScript



React JS

