단축 평가 논리 계산법

console.log(true && 'hello'); // hello

console.log(false && 'hello'); // false

console.log('hello' && 'bye'); // bye

// Truthy한 값이면 뒤에 값 표시

console.log(null && 'hello'); // null

console.log(undefined && 'hello'); // undefined

console.log('' && 'hello'); // ""

console.log(0 && 'hello'); // 0

console.log(1 && 'hello'); // hello

console.log(1 && 1); // 1

// &&

// 특정 값이 유효한 경우에만 어떤 값을 조회할 때 사용

const object = null;

const name = object && object.name;

console.log(name); // null

const object = { name : 'Jin' };

const name = object && object.name;

console.log(name); // Jin

// ||

// 어떤 값이 없을 때 다른 값을 사용할 때 사용

const namelessDog = {

name: '',

};

function getName(animal) {

const name = animal && animal.name;

return name || '이름이 없습니다';

}

const name = getName(namelessDog);

console.log(name); // 이름이 없습니다

단축 평가 논리 계산법은 리액트에서 조건부 랜더링할 때에 유용하게 사용할 수 있다.

비구조화 할당

const deepObject = {

state : {

information : {

name : 'Jiwon',

languages : ['Korean', 'English']

}

},

value: 5

}

// 비구조화 할당 두 번 하기

const { name, languages } = deepObject.state.information;

const { value } = deepObject;

const extracted = {

name,

languages,

value

};

console.log(extracted);

// 선호 되지 않는 방식

const {

state : {

information : {

name, languages

}

},

value

} = deepObject;

const extracted = {

name,

languages,

value

};

console.log(extracted);