

23114799 이영록

2주차 3회차 과제



그렇다면 암묵적 타입 변환이 무조건적으로 좋지 않은 문화이자 기능인가?

나는 좋다고 생각한다 ps하다보면 종종 숫자랑 문자열을 이리저리 바꿔야할때가 존재하는데 c언어를 사용하고있으면 암묵적 타입변환이 있는 자바로 풀생각이 든적이 있기때문이다



아래 설명에 따라 단축 평가를 이용하여 아래의 if문처럼 작동하는 true 값 여부를 판별하는 코드를 빈칸에 알맞게 작성해보시오. 결과도 내시길 바랍니다.

- 빈칸의 길이와 정답의 길이가 상이할 수 있습니다.

```
var isThereMessage = true;
var message = '';

if(isThereMessage) message = '멘토는 죽어있다.';

message = _____ ;
console.log(message);
```

```
> var a=true;
< undefined
> var message='';
< undefined
> if(a)message='멘토는 죽어있다.';
< '멘토는 죽어있다.'
> message = a&&'멘토는 죽어있다.';
< '멘토는 죽어있다.'
> console.log(message);
멘토는 죽어있다.
```

a가 true값이기에 if(a)는 작동이 되고 message에 '멘토는 죽어있다.'라는 값이 들어간다 &&의 왼쪽피연산자가 true라면 오른쪽 피연산자를 반환하고 false라면 왼쪽 피연산자를 반환함 ||의 경우 왼쪽 피연산자가 true라면 왼쪽피연산자를 반환하고 false라면 오른쪽피연산자를 반환함

아래의 코드를 실행해보시고 왜 결과 값이 그렇게 나오는지 생각해 보세요

```
var person = {
  firstName: 'turtle',
  last_name: 'park'
};

console.log(person);
//=====

var word1 = {
  var: '',
  function: ''
};

console.log(word1);
//=====

//포문짜리 키 동적 생성
var objES5 = {}
var keyES5 = 'ES5'
objES5[keyES5] = 'world';

console.log(objES5);
//=====

//제산형 포문짜리 이름
var keyES6 = 'HELL';
var objES6 = {[keyES6]: 'o'};

console.log(objES6);
//=====

var emptyObj = {
  '' : ''
};
console.log(emptyObj);
//=====

var numObj = {
  1 : 0,
  2 : 1,
  3 : 2
};
console.log(numObj);
//=====

var duplicateObj = {
  name : 'park',
  name : 'kim'
};
console.log(duplicateObj);
//=====
```

```
> var person = {
  firstName: 'turtle',
  last_name: 'park'
};
console.log(person);

▶ {firstName: 'turtle', last_name: 'park'}
```

함수를 출력하게 하면 뭘
정의하든간에 다출력한다?

```
> var word1 = {
  var: '',
  function: ''
};
console.log(word1);

▶ {var: '', function: ''}
```

이것도 동일

이거보니 생각이 좀 바뀜
c언어의 배열느낌을 다루면
될듯 배열은
0~1이지만여기서는
설정해주는대로 키값? 대로
들어가는듯함 그런데
특이하게도 키값에도 값이
들어가는듯함

```
> var objES5 = {}
var keyES5 = 'ES5'
objES5[keyES5] = 'world';
console.log(objES5);
```

```
▶ {ES5: 'world'}
```

```
< undefined
```

```
> console.log(keyES5);
```

```
ES5
```

안의 값을 바꾸는 즉시
바뀌는 그런건 아닌가봄
바꿀때마다 새로 선언을
해줘야하는듯함 변수안의
값을 나오게 하고싶으면
변수를 쓰는게 아니라
[]안에다가 넣어야 변수값이
나오는듯 함

```
> var keyES6 = 'HELL';
var objES6 = {[keyES6]: 'o'};
console.log(objES6);
```

```
▶ {HELL: 'o'}
```

```
< undefined
```

```
> keyES6 = 'HELL1';
```

```
< 'HELL1'
```

```
> console.log(objES6);
```

```
▶ {HELL: 'o'}
```

```
> objES6 = {keyES6: 'o'};
```

```
< ▶ {keyES6: 'o'}
```

```
> var emptyObj = {
  '' : ''
};
console.log(emptyObj);
```

```
▶ {"": ""}
< undefined
```

왼쪽은 키를 나타내고
오른쪽은 값을 나타내는데
공백을 나타낼때 어떻게
나타내냐?

줄바꿈은 처리가 안되는듯
따로 적어줘야하는듯?

```
> var numObj = {
  1 : 0,
  2 : 1,
  3 : 2
};
console.log(numObj);
```

```
▶ {1: 0, 2: 1, 3: 2}
```

```
> var duplicateObj = {
  name : 'park',
  name : 'kim'
};
console.log(duplicateObj);
```

```
▶ {name: 'kim'}
< undefined
```

궁금증이 여기서 풀림
name같이 왼쪽에 오는것들은
다 변수같은 취급을 해주고
변수가 다시 정의되었으니까
마지막으로 정의된 kim이
나온다.

,은 사용자가 입력해서
나온게아니라 자동으로
나오는거구나 싶다

오류가 나네요 하하

</> source code

```
1 process.stdin.resume();
2 process.stdin.setEncoding('utf8');
3
4 // your code goes here
5 var wind = {
6   'last_name' : 'park',
7   1: 10
8 };
9 wind['last_name'];
10 wind.last_name;
11 wind[last_name];
12 wind['last_name'];
13 wind.1;
14 wind.'1';
15 wind[1];
16 wind['1']
```

input Output

Runtime error #stdin #stdout #stderr 0.07s 31648KB

Runtime error #stdin #stdout #stderr 0.07s 32004KB