

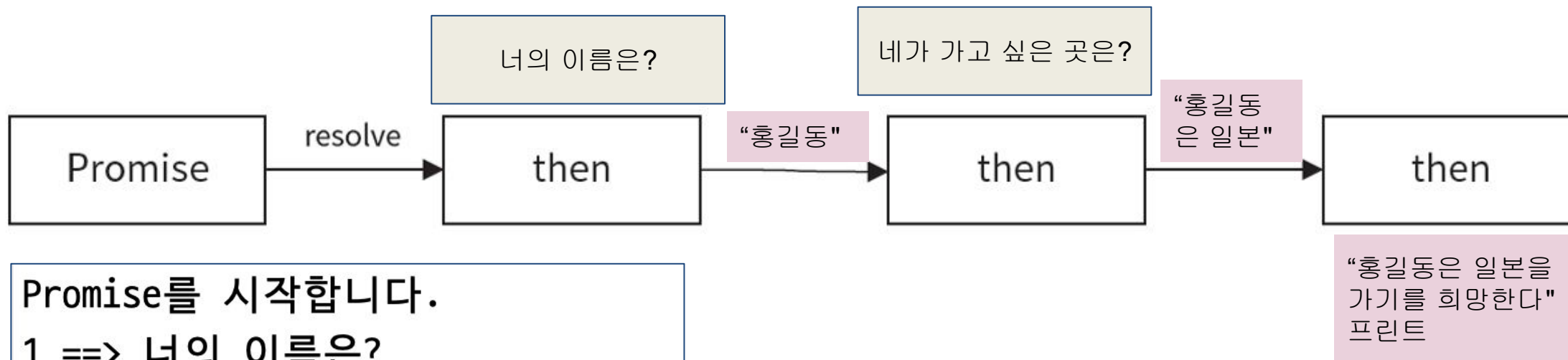
2024년 상반기 K-디지털 트레이닝

ES6 Promise + Proxy

[KB] IT's Your Life

Q1 - Promise

- 다음 그림과 같이 **Promise**를 이용해 프로그래밍 하시오.
- 에러 발생시 “에러 발생함 → 에러내용 출력”



Promise를 시작합니다.

1 ==> 너의 이름은?

2 ==> 홍길동

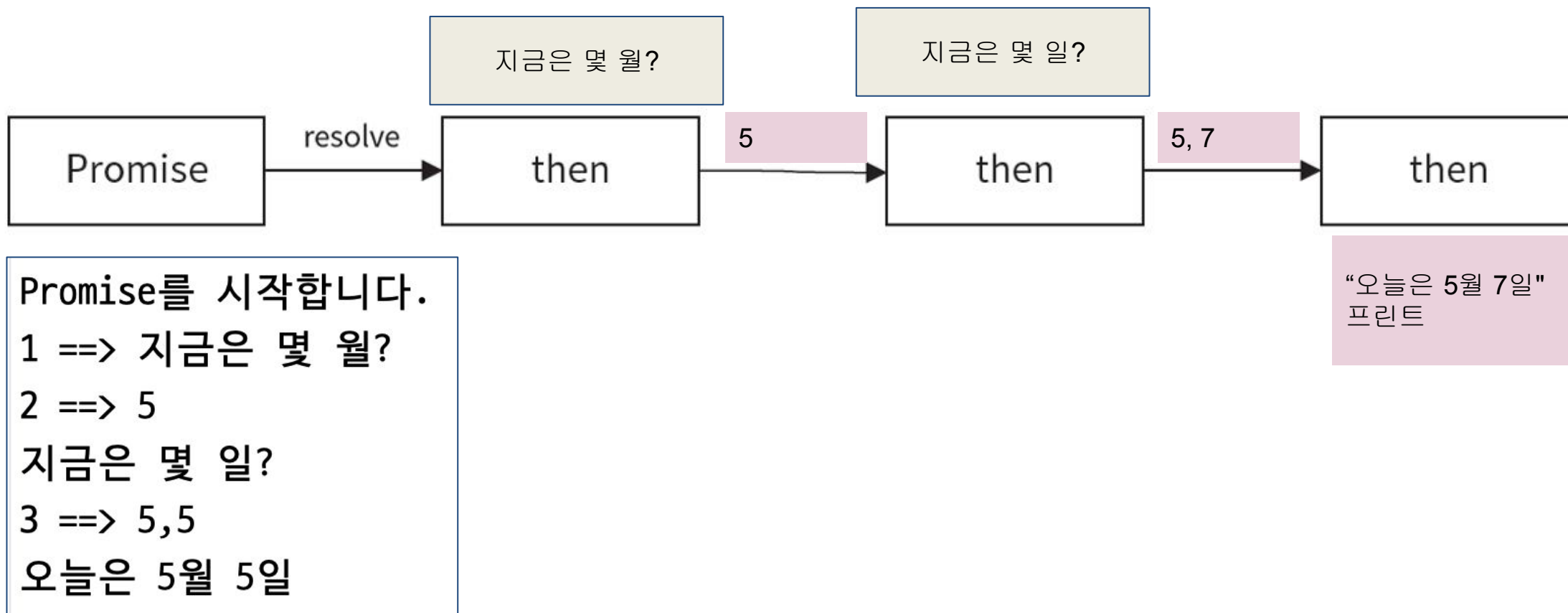
네가 가고 싶은 곳은?

3 ==> 홍길동은 일본

홍길동은 일본을 가기를 희망한다.

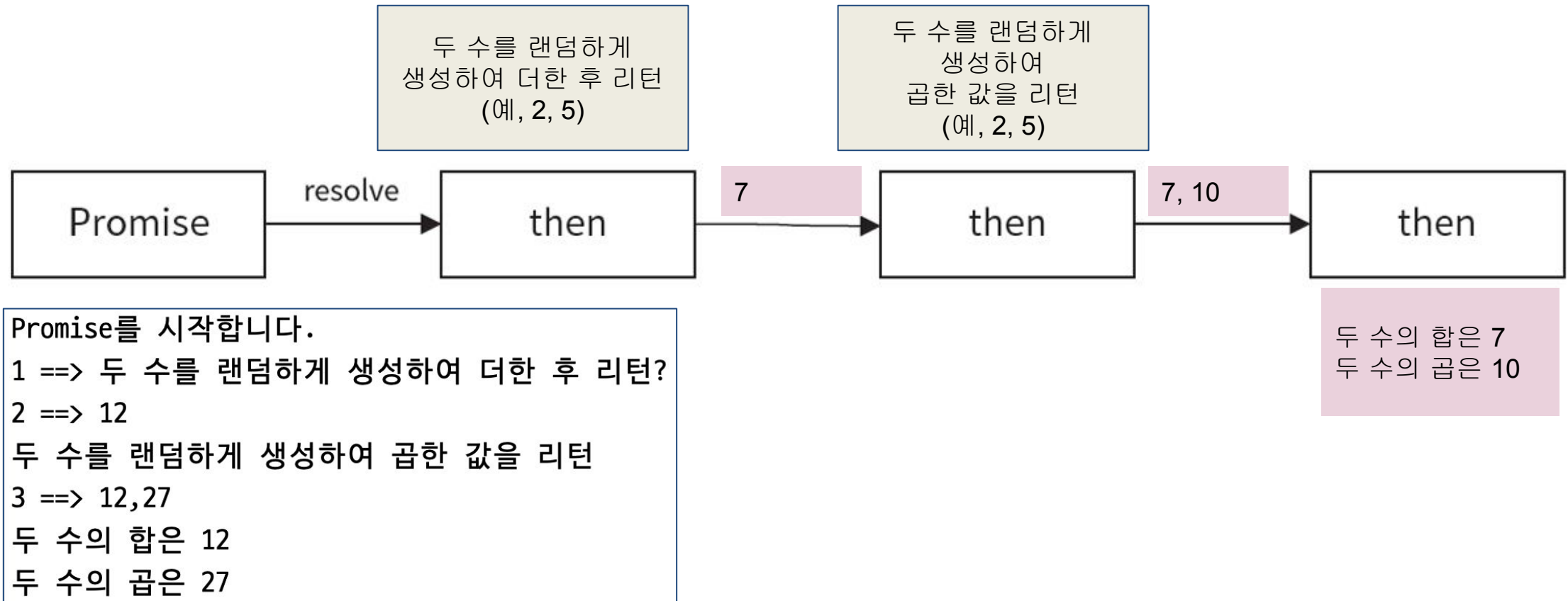
Q2 - Promise

- 다음 그림과 같이 **Promise**를 이용해 프로그래밍 하시오.
- 에러 발생시 “에러 발생함 → 에러내용 출력”
- **Date**객체 활용



Q3 - Promise

- 다음 그림과 같이 **Promise**를 이용해 프로그래밍 하시오.
- 에러 발생시 “에러 발생함 → 에러내용 출력”
- **Math.random()**활용



Q4 - 전개 연산자

- 다음의 객체를 복사하여, **email**을 추가 후 새로운 객체를 생성 후 프린트하시오.
- 새롭게 생성된 객체의 **age**값을 **100**으로 수정후 전체 객체의 내용 프린트하시오.
- 새롭게 생성된 객체와 기존의 객체가 동일한지 비교하여 프린트하시오.

```
let obj1 = { name : “박문수”, age : 29 };
```

```
obj1>> { name: '박문수', age: 29 }  
obj2>> { name: '박문수', age: 29, email: 'winner@email.com' }  
-----  
obj1>> { name: '박문수', age: 29 }  
obj2>> { name: '박문수', age: 100, email: 'winner@email.com' }  
false
```

- 다음을 입력받아 배열에 넣은 후
- 각각 동일한 값이 몇 개인지 세서 프린트하시오.

- 입력 : 맑음, 흐림, 흐림, 보통, 보통, 흐림
- 출력 :
 - 전체 출력 : 맑음, 흐림, 흐림, 보통, 보통, 흐림
 - 개수 출력 : 맑은 - 1, 흐림 - 3, 보통 - 2

```
일주일간 날씨를 입력하세요.>> 맑음,흐림,흐림,보통,보통,흐림
전체 날씨 출력>> 맑음,흐림,흐림,보통,보통,흐림!
[ '맑음', '흐림', '흐림', '보통', '보통', '흐림' ]
one>> 맑음
has(key)>> false
one>> 흐림
has(key)>> false
one>> 흐림
has(key)>> true
one>> 보통
has(key)>> false
one>> 보통
has(key)>> true
one>> 흐림
has(key)>> true
맑음 - 1
흐림 - 3
보통 - 2
```

- terminal에서 입력 예제

```
// 새로운 인터페이스 생성
import readline from 'readline';

const rl = readline.createInterface({
  input: process.stdin, // 표준 입력 스트림 사용
  output: process.stdout // 표준 출력 스트림 사용
});

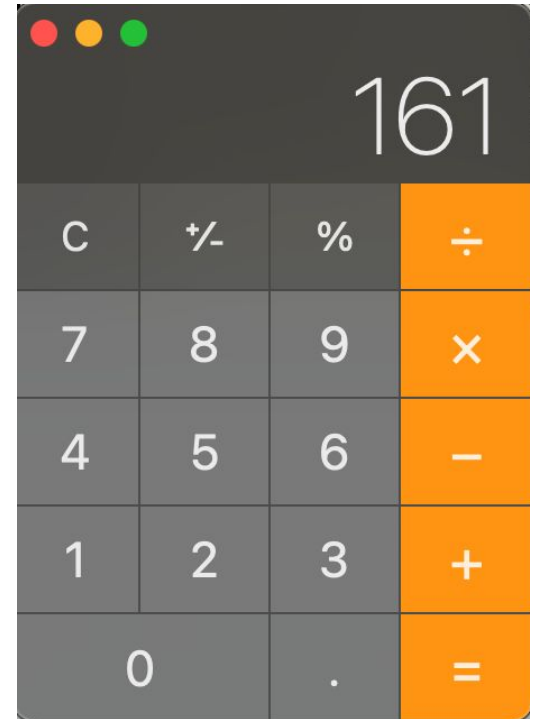
// 사용자에게 메시지를 출력하고 입력을 기다림
rl.question('이름을 입력하세요: ', (name) : void => {
  console.log(`안녕하세요, ${name}님!`);
  rl.close(); // readline 인터페이스 종료
});
```

이름을 입력하세요: hong
안녕하세요, hong님!

- 다음을 입력받아 배열에 넣은 후
- 각각 동일한 값이 몇 개인지 세서 프린트하시오.

- 입력 : 11,12,15,89,34
- 처리 : 공백 양쪽에 4개씩 있음. 공백 제거 후 각각 배열에 넣음.

값 5개를 입력>> 11,12,15,89,34
입력받은 값>> 11,12,15,89,34
공백 제거 전 글자수 >> 26
공백 제거 후 글자수 >> 14
전체 합은 >> 161
정수변환 후 정렬된 배열>> 11,12,15,34,89
제일 큰 수와 제일 작은 수의 차는>> 78



- 다음과 같은 객체에 값을 변경했을 때와 추출되었을 때 함수를 Proxy객체를 이용해 다음과 같이 출력되도록 프로그래밍하시오.

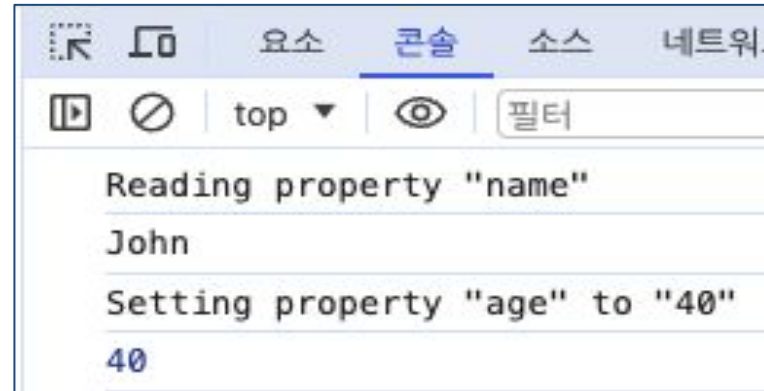
```
let obj = { name : “홍길동”, age : 20 };
```

```
console.log(proxy.name);    //읽기 작업 get 호출  
proxy.name = “hahahaha”;    //쓰기 작업 set 호출  
proxy.age = 30;             //쓰기 작업 set 호출
```

```
## get name  
홍길동  
## set name  
## set age
```

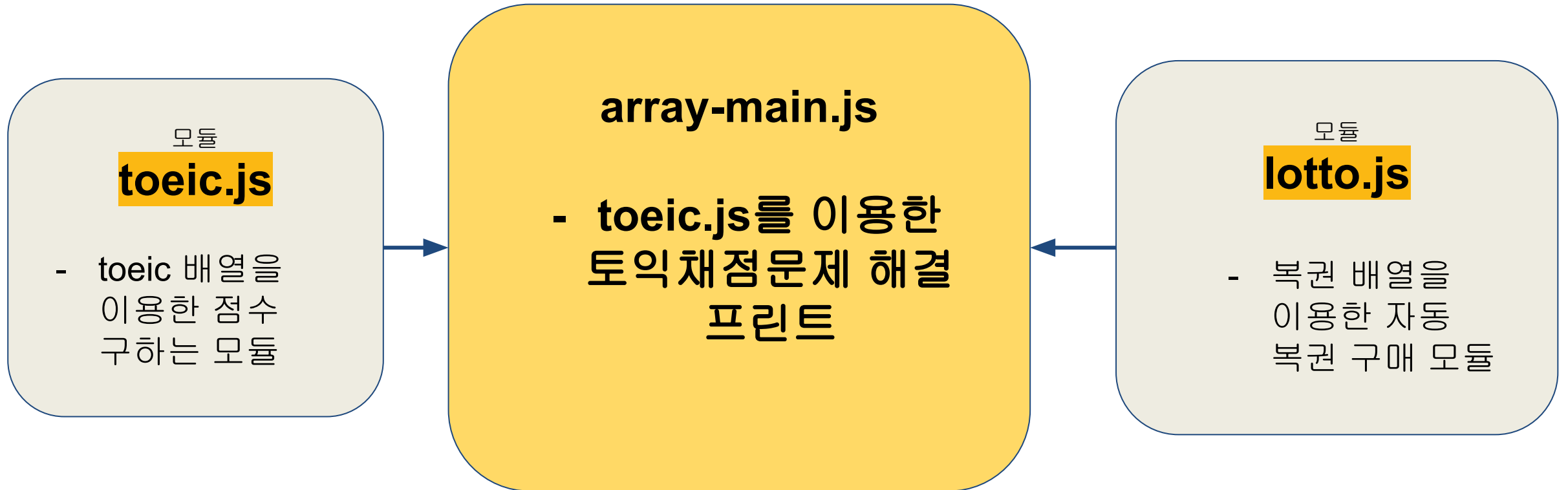
- 다음과 같은 객체에 값을 변경했을 때와 추출되었을 때 함수를 Proxy 객체를 이용해 다음과 같이 출력되도록 프로그래밍하시오.

```
const target = {  
  name : 'John',  
  age : 30  
}
```



```
## get phone  
존재하지 않는 속성 phone입니다.  
undefined  
## get name  
John  
## set age  
40
```

- 다음 그림과 같이 구현하고, 실행결과를 출력하시오.



- 토익문제

- 1차원 배열 2개를 이용하여 다음 토익 채점 문제를 해결하시오.
- toeic 답안을 Math.random()을 이용하여 0~999번까지 생성하여 배열에 넣으시오.
- toeic 내답안을 Math.random()을 이용하여 0~999번까지 생성하여 배열에 넣으시오.
- 답의 범위는 1~4까지 발생하시오. (문제 1개당 점수는 1점)
- 두 배열을 비교하여 채점 후 toeic점수를 프린트하시오.

- 복권문제

- 1차원 배열에 6개의 랜덤한 복권번호를 만들어 넣으시오.
- 복권의 번호는 유일해야함.
- Math.random()을 이용하여 1~45번 사이의 값을 넣으시오.
- 전체 복권 정보를 프린트하시오.

- 출력예시

복권번호: [41, 37, 13, 24, 30, 42]

TOEIC 점수: 237점

984>> 1 2 --> 오답
985>> 3 1 --> 오답
986>> 1 2 --> 오답
987>> 4 3 --> 오답
988>> 2 2 --> 정답
989>> 3 1 --> 오답
990>> 3 2 --> 오답
991>> 1 1 --> 정답
992>> 2 3 --> 오답
993>> 2 1 --> 오답
994>> 3 1 --> 오답
995>> 3 4 --> 오답
996>> 3 4 --> 오답
997>> 1 1 --> 정답
998>> 1 3 --> 오답
999>> 4 1 --> 오답
TOEIC 점수: 248점

- 출력예시

문항	답안	내답
0	1	1
1	2	3
2	1	2
3	4	4
- - - - -		
999	1	4

- 오답 개수 : 300개
- 정답 개수 : 700개
- 내 toeic점수 : 700점

대박복권 이번주 번호
- 41, 1, 6, 34, 22, 5

수고하셨습니다!

