

자바스크립트 문제 풀이 안내 (공통)

1. 가장 큰 값 구하기

함수의 이름은 `maxValue` 입니다.
인자로써 정수들이 들어있는 배열을 파라미터로 전달받습니다.
배열로 전달받은 숫자 중 가장 큰 숫자와
숫자를 내림차순으로 정렬된 배열을 객체 형태로 반환해야 합니다.

ex)

```
const a = maxValue([1,2,3,4,5])
console.log(a)
```

결과값

```
{
  maxValue: 5,
  sortArr = [5,4,3,2,1]
}
```

2. 편의점 물건 확인하기

당신은 편의점에서 야간 아르바이트를 하는 학생입니다
당신이 일하는 시간에 편의점에 채워놓을 물건이 들어오며
당신은 재고의 개수와 물품이 맞는지 확인해야 합니다.

일의 순서는 아래와 같습니다.
1. 당신은 아래와 같은 전산표를 전달받습니다.

품목	갯수
야채곰창	5
바나나우유	10
삼각김밥	15
도시락	10
샌드위치	10

2. 당신은 물품을 세어본 후 해당 제품이 표의 갯수와 같은지 확인해야 합니다.

최종목표

- 물품명과 갯수를 파라미터로 전달받는 함수를 만듭니다.
함수의 이름은 마음대로 하셔도 괜찮습니다.
- 만약 표의 값과 갯수가 일치하지 않는다면
"전산표와 일치하지 않습니다"를 로그 합니다.

같다면

“전산표와 일치합니다”를 로그 합니다.

3. 전산표에 없는 물건이라면 확인을 위해
해당 물건의 이름을 콘솔에 로그 합니다.

ex) 함수명(물품명, 갯수);

결과값

”전산표와 일치합니다.“ (일치)

”전산표와 일치하지 않습니다.“ (불일치)

”바사삭치킨“ (품목 없음)

3. 배열 다루기

당신이 구현하고자 하는 함수는
배열과 정수를 파라미터로 전달 받습니다.

함수 이름은 정해지지 않았으니 마음대로 하셔도 괜찮습니다.

전달받은 배열은 정확히 각 요소마다 값을 10을 더 합니다.

이후 값이 10씩 더해진 배열에서, 나머지 파라미터로 전달받은 정수가 있는지 확인하여

값이 없다면 콘솔에 검사하고자 하는 값이 없다는 로그를 보인 후 함수를 종료하세요
값이 있다면 해당 숫자를 제외한 배열을 반환하는 함수를 만드세요.

최종 목표

1. 해당 함수는 배열과 검사하고자 하는 수를 파라미터로 전달 받습니다.
2. 해당 함수는 배열의 각 요소에 10을 더한 후 검사하고자 하는 수가 있는지 검사합니다.
3. 만약 값이 없다면 결과값이 없음을 콘솔에 로그합니다.
4. 값이 있다면 검사하고자하는 수를 제외한 배열을 반환합니다.
5. 값이 없을 때 console.log에 undefined가 찍혀서는 안됩니다.

ex)

```
const arr = [1,2,3,4,5]
const a =함수명(arr, 11)
console.log (a)
```

결과값

[12,13,14,15] (값 있음)

”결과값이 없습니다“ (값 없음)

4. 짝수와 홀수의 개수 구하기

숫자로 된 배열을 파라미터로 받는 함수 solution이 있습니다.

이 솔루션은 파라미터로 전달받은 함수에서 짝수와 홀수의 개수를 배열로 반환합니다.

최종목표

1. 함수 solution을 숫자 배열을 파라미터로 받는다.
2. 배열 중 짝수와 홀수를 각각 구분하여 길이를 구한다.
3. 길이를 구한 후 각각의 개수를 요소로 담은 배열을 반환한다.
단, 요소의 순서는 짝수, 홀수 순이다.

ex)

```
const arr = [1,2,3,4,5]
const a = solution(arr)
console.log(a)
```

결과값

[2,3]

[짝수의 개수, 홀수의 개수]

5. 심화문제 (선택)

심화 문제의 올바른 풀이를 위해서는 아직 배우지 않은 개념도 이해하고 있어야 합니다.. (클로저, this)

그러나 반드시 올바르게 문제를 풀이하지 않아도 괜찮으니 아래와 같은 기능을 구현해보세요.

올바른 풀이가 아닌 현재까지 배운 개념만으로도 기능 작동에는 문제가 없는 것을 확인하였습니다.

아래와 같은 기능을 하는 오브젝트를 반환하는 함수를 만드세요

파라미터로는 최종 목표지 까지와의 거리를 전달 받습니다. 함수 이름은 car입니다.

car는 각각 시동걸기, 시동끄기, 주행 기능을 가지고 있습니다.

해당 자동차는 연식이 오래되어 최대 40km밖에 운전하지 못합니다.

40km 이상 주행하려고 한다면 안전 사고 위험이 있어 강제로 시동을 종료합니다.

주행거리까지 1km 단위로 상황판(콘솔)에 나타냅니다.

최종 목표

1. 시동이 걸리면 시동이 걸렸다는 메시지가 콘솔에 로그됩니다.
 2. 시동이 꺼지면 시동이 꺼졌다는 메시지가 콘솔에 로그됩니다.
 3. 주행 중이면 최종 목표 거리까지 1km 단위로 콘솔에 거리가 로그됩니다.
 4. 최종 목표거리에 도달하면 주행이 완료되었다는 메시지가 콘솔에 로그됩니다.
 5. 최종 목표거리가 40km 이상인 상태에서 주행을 시도하면
안전 위험으로 시동을 종료했다는 메시지가 콘솔과 로그됨과 동시에 시동이 꺼집니다.
 6. 시동이 걸리지 않으면 주행을 할 수 없습니다.
 7. 시동이 걸려있는 상태에서 시동을 다시 걸 수 없습니다.
 8. 시동이 꺼져있는 상태에서 시동을 다시 끌 수 없습니다.
-

6. 문제 만들기

문제 만들기

위와 같은 문제를 예시로 이제는 직접 문제를 만들어보는 것입니다.

동료들과 이야기하여 문제를 만들어보세요 :)

최종 목표

1. 만들고 싶은 기능 정하기
2. 필요한 인풋(입력값)과 아웃풋(출력값) 정하기
3. 인풋과 아웃풋을 토대로 기능 구현하기
4. 문제 형태로 만들어 제출해주세요.
 메일로 제출해주시면 됩니다:)
