



08.23_실습 강의 자료

0교시

강의 이해도 체크

1교시

1-01 합격 여부 확인하기

▼ if문, else if문, else문 사용방법은?

if, else if, else 사용방법을 실습해보겠습니다.

python의 elif가 else if로 바뀐것 말고는 동일합니다.

주어진 지시사항을 순서대로 따라가 보면

먼저 if문으로 세 과목의 점수를 각각 확인하고 점수가 40점 보다 낮은 과목이 있는지 체크합니다.

```
if (scores.kor < 40 || scores.mat < 40 || scores.eng < 40) {  
    return "fail";  
}
```

다음으로 세 과목의 평균을 구해 60점 보다 낮은지 체크합니다.

```
else if ((scores.kor + scores.mat + scores.eng) / 3 < 60) {  
    return "fail";  
}
```

마지막으로 위의 두 경우가 아니라면 각각 점수가 40점 이상이고 평균이 60점 이상이므로 pass를 리턴하는 else문을 작성합니다.

```
else {  
    return "pass";  
}
```

▼ 다른 풀이 방법은?

조금 더 효율적으로 푸는 방법은?

| 함수로 만들기

함수로 만들게 되면 재사용하기 수월하므로 다른 곳에도 적용할 수 있습니다.

```
function result() {  
  if (scores.kor < 40 || scores.mat < 40 || scores.eng < 40) {  
    return "fail";  
  } else if ((scores.kor + scores.mat + scores.eng) / 3 < 60) {  
    return "fail";  
  } else {  
    return "pass";  
  }  
}  
  
document.write(result());
```

| Object.values 사용하기

Object.values()를 사용하면 객체의 value값들 불러올 수 있습니다.

```
const scoreArr = Object.values(scores);  
console.log(scoreArr); // [ 55, 75, 50 ]
```

| Array.reduce() 사용하기

Array.reduce()를 사용해서 array안의 값들을 모두 더해줄 수 있습니다.

```
const avg = scoreArr.reduce((total, num) => total + num) / 3;  
console.log(avg); // 60
```

reduce 함수는 배열의 각 요소에 대해 주어진 리듀서(reducer)함수를 콜백함수로 실행하고 하나의 결과값을 반환합니다.

reduce()는 2개의 인자값을 받습니다.

- `callbackfn` : 배열의 각 요소에 실행할 리듀서 함수
- `initialValue` : 리듀서 함수에 첫번째 누적값으로 제공할 초기 값(없으면 배열의 첫번째 요소가 초기값이 됨)

리듀서함수는 4개의 인자값을 받습니다.

- `accumulator` : 누적값
- `currentValue` : 현재 요소값
- `currentIndex` : 현재 요소값의 index
- `array` : `reduce()`를 호출한 배열

```
const arr = [1, 2, 3, 4, 5];

console.log("초기값 없음");
arr.reduce((total, now, nowIdx, arr) => {
  console.log("total :", total);
  console.log("now :", now);
  console.log("nowIdx :", nowIdx);
  console.log("arr :", arr);
  console.log("\n");

  return total + now;
});

console.log("\n초기값 있음");
arr.reduce((total, now, nowIdx, arr) => {
  console.log("total :", total);
  console.log("now :", now);
  console.log("nowIdx :", nowIdx);
  console.log("arr :", arr);
  console.log("\n");

  return total + now;
}, 0);
```

Array.find() 사용하기

`Array.find()`를 사용하면 조건을 만족하는 첫번째 요소의 값을 반환할 수 있습니다.

```
console.log(scoreArr.find((item) => item < 40));
// undefined
```

filter 사용해보기

Array.filter()를 사용하면 조건을 만족하는 요소들만 새로운 배열에 담아 반환할 수 있습니다.

```
list.filter(v => v < 40)
// 40이하인 요소들만 있는 배열 반환
```

2교시

1-02 짝수 판별기

▼ 반복문 사용방법은?

for ([초기문]; [조건문]; [증감문])

```
for (let i = 2; i <= 100; i = i + 2) {
  document.write(i);
}
```

while (조건문)

```
let num = 2;
while (num <= 100) {
  document.write(num);
  num += 2;
}
```

▼ for문과 while문 사용기준은?

정해진 답은 없지만 보통

반복횟수가 정해졌을 때는 for문을

종료조건이 정해졌을 때는 while문을 을 많이 사용합니다 😊

1-03 점심 메뉴 찾기

▼ for문의 여러가지 풀이법은?

지시사항을 읽고 어떻게 코드를 작성해야 할 지 생각해보세요!

방법은 하나가 아닙니다!

▼ 다른 방법은?

Array.map() 사용하기

map은 배열의 각각의 요소를 한번씩 순서대로 불러 반환값으로 새로운 배열을 생성합니다.

각 요소를 콜백함수에 인자로 넣어 반환값을 새로운 배열로 반환하죠

```
const arr = ["월요일", "화요일", "수요일", "목요일", "금요일"];

// map
const map = arr.map((item) => "출근하는 " + item);
console.log(map);
/*
"출근하는 월요일",
"출근하는 화요일",
"출근하는 수요일",
"출근하는 목요일",
"출근하는 금요일"
*/
```

3교시

1-04 B로 시작하는 과일을 찾아줘

▼ b로 시작하는 과일을 어떻게 찾죠..?

fruits[0]에 해당하는 apple을 골랐다면

단어 중 첫번째 글자가 b인지를 판별하려면 어떻게 해야할까요?

JS에는 string안의 글자들도 index로 접근할 수 있죠~!

예를들어 if(apple[0] === 'b')인지를 보면 되겠죠?

그럼 어떤 단어를 선택할 지는 어떻게 해야할까요?

단어를 선택하는 방법은 array의 index를 통해서 할 수 있죠~!

예를들어 fruits[0]은 apple이 됩니다.

이 두가지 과정을 합치면 if(fruits[0][0] === 'b')가 됩니다.

fruits의 모든 단어를 검사하려면 어떻게 해야할까요?

```
for (let i = 0; i < fruits.length; i++) {
  if (fruits[i][0] === "b") {
    // code
  }
}
```

이렇게 하면 되겠죠??

▼ 다른 풀이

map 사용한 풀이

```
fruits.map((item) => {  
  if (item[0] === "b") sum++;  
});  
console.log(sum);
```

filter사용한 풀이

```
console.log(fruits.filter((item) => item[0] === "b"));  
  
sum = fruits.filter((item) => item[0] === "b").length;  
  
console.log(`b로 시작하는 과일은 ${sum}개`);
```

map과 filter가 비슷해보이지 않으신가요??

map과 filter의 차이점에서 대해서 한번 알아보겠습니다.

map은 각 요소를 콜백함수에 인자로 넣어 반환값을 새로운 배열로 반환하는 반면

filter는 각 요소들중 주어진 조건에 만족하는 값만을 새로운 배열로 반환합니다.

```
const arr = ["월요일", "화요일", "수요일", "목요일", "금요일"];  
  
// map  
const map1 = arr.map( (_, idx) => idx % 2 !== 0 );  
console.log(map1);  
/* [ false, true, false, true, false ] */  
  
const map2 = arr.map((item) => "출근하는 " + item);  
console.log(map2);  
/* [ '월요일', '화요일', '수요일', '목요일', '금요일' ] */  
  
// filter  
const filter = arr.filter( (_, idx) => idx % 2 !== 0 );  
console.log(filter);  
/* [ '화요일', '목요일' ] */
```

| for ... of를 사용한 풀이

for ... of를 사용하면 forEach문을 사용한 것처럼 배열 안에 있는 요소를 하나씩 가져와 반복문을 실행합니다.

```
for (const v of fruits) {  
  if (v[0] === 'b') sum++;  
}
```

이렇게 모든 요소의 첫번째 값이 b인지를 체크 할 수도 있죠 😊

1-05 특정 학생 정보 바꾸기

▼ object안의 값 다루는 방법

4번문제 에서 한 것 처럼 반복문을 통해서 array안의 object안의 값들을 하나씩 확인합니다.

```
for (let i = 0; i < students.length; i++) {  
  // code  
}
```

object에 접근했다면 object안에서 name이 elice인 object를 찾아야 겠죠?

```
for (let i = 0; i < students.length; i++) {  
  if (students[i].name === "elice") {  
    // code  
  }  
}
```

name이 elice인 object를 찾았다면 그 object의 name을 computer science바꿔줍니다.

```
for (let i = 0; i < students.length; i++) {  
  if (students[i].name === "elice") students[i].major = "computer science";  
}
```

▼ 다른 풀이는?

map과 arrow function을 사용해서 풀면 조금 더 가독성 높게 풀 수 있습니다.

```
students.map((item) => {  
  if (item.name === "elice") item.major = "computer science";  
});
```

이처럼 우리가 원하는 값을 만들기 위해서 접근하는 방법은 매우 다양합니다.

지금까지 배운 것을 토대로 여러분의 창의적인 풀이를 생각해보세요 😊

추가!!

▼ 향상된 for 문

forEach 문

array에서만 사용가능한 메서드로 원소들을 나열해서 검사하는 기능이 있습니다.

```
const arr = ["엘리스", "AI", "9기", "레이서 분들", " 화이팅"]

for(let i=0; i< arr.length; i++){
  console.log(item)
}

arr.forEach(item => {
  console.log(item)
});
```

https://developer.mozilla.org/ko/docs/Web/JavaScript/Reference/Global_Objects/Array/forEach

for ... in 문

객체에 사용하는 for문으로 객체 안에 있는 프로퍼티들을 가지고 반복적인 동작들을 수행할 때 사용합니다.

객체의 key값에 바로 접근할 수 있기 때문에 일반적인 for문으로는 대체할 수 없는 특별한 반복문입니다.

객체 내부의 프로퍼티값들을 하나씩 다뤄야 하는 경우에 유용하게 사용할 수 있습니다.

```
let obj = {
  name : "엘리스 AI",
  age : 9,
  jobs : "레이서",
  track : "엔지니어"
}

for(let key in obj){
  console.log(key)
  console.log(obj[key])
}
```

이때 주의할 점은 객체는 정수형 프로퍼티 네임을 오름차순으로 먼저 정렬하고, 나머지 프로퍼티들은 추가한 순서대로 정렬하는 특징이 있기 때문에 순서에 유의해서 사용해야 합니다.

for ...of 문

배열을 다루는 반복문으로 배열안에 값들을 변수로받아 사용합니다 forEach와 마찬가지로 배열의 길이만큼 자동으로 실행하기 때문에 간략하게 사용할 수 있습니다.

```
for (let item of arr) {  
  console.log(item);  
}
```

사실 배열도 객체이기 때문에 for in에 배열도 사용할 수 있지만 for in문은 객체에 최적화 되어있다보니 실행하는 환경에따라 배열에 있는 length프로퍼티 같은 것들이 변수로 할당될 가능성이 있어 배열에는 for in문을 사용하지 않기를 권장합니다.

배열의 index를 활용하고 싶다면 그냥 for문을 사용하길 권장드립니다.