

좋은 프로그램은

- 가독성
- 재사용성
 - 함수 : 반복되는 코드는 함수로 만들어서 사용한다.
 - 객체 : 데이터(변수)와 함수를 묶어서 재사용할 수 있도록 만든다.

의사코드(pseudo-code)

- 의사(疑似: 비교할 의, 비슷할 사 | Pseudo: 가짜의-)
- 컴퓨터 프로그램이 수행해야할 내용을 우리가 사용하는 언어로 간략히 서술해 놓은 것
- 장점
 - 코딩입력을 시작하기 전, **사고를 좀 더 명확히 정립**하게 만들어 주어 프로그램을 설계하는데 도움
 - **코드검토**(code review)가 더 쉬워짐 : 소스코드의 실행없이 상세 설계를 검토
 - **코드 수정이 용이**하게 만듦 : 전체의 코드를 수정하는 것보다는 의사코드 몇줄 수정하는 것이 쉽다. 미리 오류 수정이 용이
 - **코멘트**(주석) 작성에 대한 부담을 덜어준다. 코딩 후에 따로 의사코드를 작성할 필요없이 의사코드 자체자 코멘트가 된다.

피겨스케이팅 점수 계산

• 요구사항

피겨점수는 기술점수와 예술점수를 더한 점수입니다. 기술점수는 심사위원들의 점수 중에서 최고점수와 최저점수를 제외한 산술평균으로 계산한다.

• 수도코드

1. 기술점수 계산

1. 최댓값을 계산
2. 최솟값을 계산
3. 합계를 계산
4. 합계에서 최댓값과 최솟값을 뺀다
5. 평균 계산 : $\text{합계} / (\text{심사위원수} - 2)$

2. 예술점수 계산

3. 기술점수와 예술점수 합계

• 수도코드

1. 기술점수 계산

1. 최댓값을 계산
2. 최솟값을 계산
3. 최대와 최소값을 제외한 배열을 새로 만들
4. 합계 계산
5. 평균 계산 : $\text{합계} / \text{심사위원수}$

2. 예술점수 계산

3. 기술점수와 예술점수 합계

//기술점수

```
let tes_score = [3,2,5,1];
```

```
let tes_max = tes_score[0];
for(let i=1; i<tes_score.length; i++){
  if(tes_score[i] > tes_max) {
    tes_max = tes_score[i];
  }
}
console.log(tes_max)
```

//구성점수

```
let pcs_score = [2,3,4,1];
```

```
let pcs_max = pcs_score[0];
for(let i=1; i<pcs_score.length; i++){
  if(pcs_score[i] > pcs_max) {
    pcs_max = pcs_score[i];
  }
}
console.log(pcs_max)
```

//기술점수

```
let tes_score = [3, 2, 5, 1];
let tes_max = max(tes_score);
console.log(tes_max);
```

//구성점수

```
let pcs_score = [2, 3, 4, 1];
let pcs_max = max(pcs_score);
console.log(pcs_max)
```



```
function max(score){
  let tes_max = tes_score[0];
  for(let i=1; i<tes_score.length; i++){
    if(tes_score[i] > tes_max) {
      tes_max = tes_score[i];
    }
  }
}
```

```
let tes_score = {
  score: [3, 2, 5, 1],
  max: function () {
    this.maxScore = this.score[0];
    for (let i = 1; i < this.score.length; i++) {
      ...
    }
    return this.maxScore;
  }
};
tes_score.max()
console.log(tes_score.maxScore)

let pcs_score = {
  score: [2, 3, 4, 1],
  max: function () {
    this.maxScore = this.score[0];
    for (let i = 1; i < this.score.length; i++) {
      ...
    }
    return this.maxScore;
  }
};
pcs_score.max()
console.log(pcs_score.maxScore)
```

```
let tes_score = new FigureCalc([3,2,5,1]);
tes_score.max()
console.log(tes_score.maxScore)

let pcs_score = new FigureCalc([4,3,2,5,1]);
pcs_score.max()
console.log(pcs_score.maxScore)

function FigureCalc(score) {
  this.score = score;
  function max(){
    this.maxScore = this.score[0];
    for(let i=1; i<this.score.length; i++){
      if(this.score[i] > this.maxScore) {
        this.maxScore = this.score[i];
      }
    }
    return this.maxScore;
  }
}
```

자바스크립트 변수 메모리 참조

- mutable 객체와 immutable 객체