

Day-17

Floatin-point Comparison

부동 소수점 비교

- 문제점

```
cout << (0.11f + 0.11f < 0.22f); // print true!!  
cout << (0.1f + 0.1f > 0.2f); // print true!!
```

- 절대 오차 범위를 사용하지 말 것

```
bool areFloatNearlyEqual(float a, float b) {  
    if (std::abs(a - b) < epsilon); // 엡실론은 사용자가 정  
        return true;  
    return false;  
}
```

- 엡실론이 “작게 보이지만” 비교되는 숫자가 매우 작을 때 너무 커질 수 있는 문제 수
정
- 비교되는 숫자가 매우 큰 경우, 엡실론이 가장 작은 반올림 오차보다 작아져 비교가
항상 false를 반환할 수 있음

부동 소수점을 꼭 비교해야 할 일이 생긴다면?

- C++의 경우 부동 소수점의 정확도를 가지고 비교하는 방법을 사용함

```
#include <iostream> //in C  
#include <cmath> // in C++  
  
#define EPSILON 0.00001 //정확도.
```

```
bool float_compare(float a, float b)
{
    return fabs(a-b) < EPSILON; // 앞서 정의한 0.00001 이하는 비교함수
```