

## 1. 출력되는 값은 무엇입니까?

정답: 보기 3번

```
String s = null;
if (s != null || !s.isEmpty()) {
    System.out.println("Answer A");
} else {
    System.out.println("Answer B");
}
```

1. Answer A
2. Answer B
3. 오류 발생

`s` 는 `null` 이기 때문에, `if` 문의 첫 번째 조건 `s != null` 의 값은 `false` 입니다.

앞쪽 메소드가 `false` 를 리턴하면 뒤쪽 메소드도 실행됩니다. 그런데 `s` 가 `null` 인 상황에서 `s.isEmpty()` 는 `NullPointerException` 에러를 일으킵니다!

## 2. 출력되는 값은 무엇입니까?

정답: 보기 2번

```
String[] strings = new String[5];
strings[0] = "Hello";
strings[2] = "";
strings[4] = "Java";

int countA = 0, countB = 0;
for (String str : strings) {
    if (str != null && !str.isEmpty()) {
        countA++;
    } else {
        countB++;
    }
}
System.out.println(countB);
```

1. 2
2. 3
3. 4
4. 5
5. 오류 발생

if문이 첫 번째 조건부터 보지만 `str != null` 은 위스가 `null` 이지 체크하는 조건입니다. `null` 이 아닌

이런 경우 두 번째 조건문도 실행하지 않아요. `str != null` 이 참이기 때문에 `str.isEmpty()` 실행하지 않아요. `str` 가 `null` 이 아니라면 `true` 를 리턴하겠죠.

두 번째 조건 `!str.isEmpty()` 은 문자열 `str` 이 빈 문자열인지 확인합니다. `strings[2].isEmpty()` 의 경우 `true` 를 리턴하겠죠.

그런데, 두 조건이 and 연산(`&&`)으로 연결되었습니다. 쏜서킷 연산에 의해 첫 번째 조건이 `true` 인 경우, 즉 `str` 이 `null` 이 아닌 경우에만 두 번째 조건을 수행하겠죠? 그렇기 때문에 앞 문제에서처럼 `NullPointerException` 은 발생하지 않습니다! 그러니 5번은 답이 아닙니다.

자, 두 조건문을 연결해서 생각해보면, 이 if문은:

1. `str` 이 `null` 이 아니면서 빈 문자열도 아닐 경우 `countA` 를 하나 증가시키고
2. `str` 이 `null` 이거나 빈 문자열(`""`)인 경우 `countB` 를 하나 증가시킵니다

그러므로 `null` 인 `strings[1]`, `strings[3]` 의 경우, 그리고 빈 문자열인 `strings[2]` 의 경우 `countB` 가 증가 되어, 최종적으로 3 이 출력됩니다.



수업을 완료하셨으면 체크해주세요.



수강생 Q&A 보기



[/questions?](#)  
질문하기

[assignment\\_id=459&sort\\_by=popular\)](#)  
[/questions/new?](#)

[assignment\\_id=459&op1=%EA%B0%9D%EC%B2%B4+%EC%A7](#)

< 이전 강의  
null 퀴즈 [\(/assignments/458\)](#)

다음 강의 > [\(/assignments/351\)](#)  
final