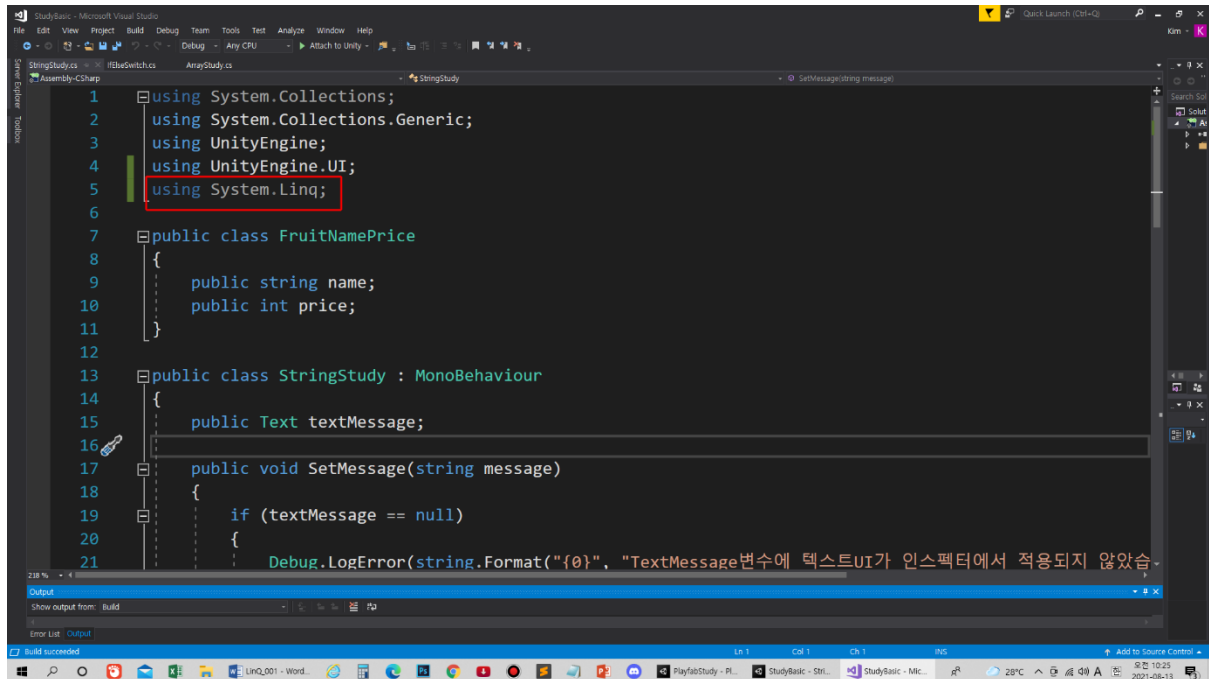


이번시간에는 LinQ구문에 대해서 알아보도록 하겠습니다. LinQ구문을 데이터베이스쪽에서 와서 조건검사를 하는데 무척 효율적입니다. 배열이나 리스트, 딕셔너리 타입으로 되어 있는 컬렉션에서 반복문을 돌지 않고도 찾고자 하는 요소들을 찾을 수 있습니다.

라이브러리를 using해 주도록 합니다.



SQL에서 가장 많이 사용하는 문법은 다음 4가지 입니다.

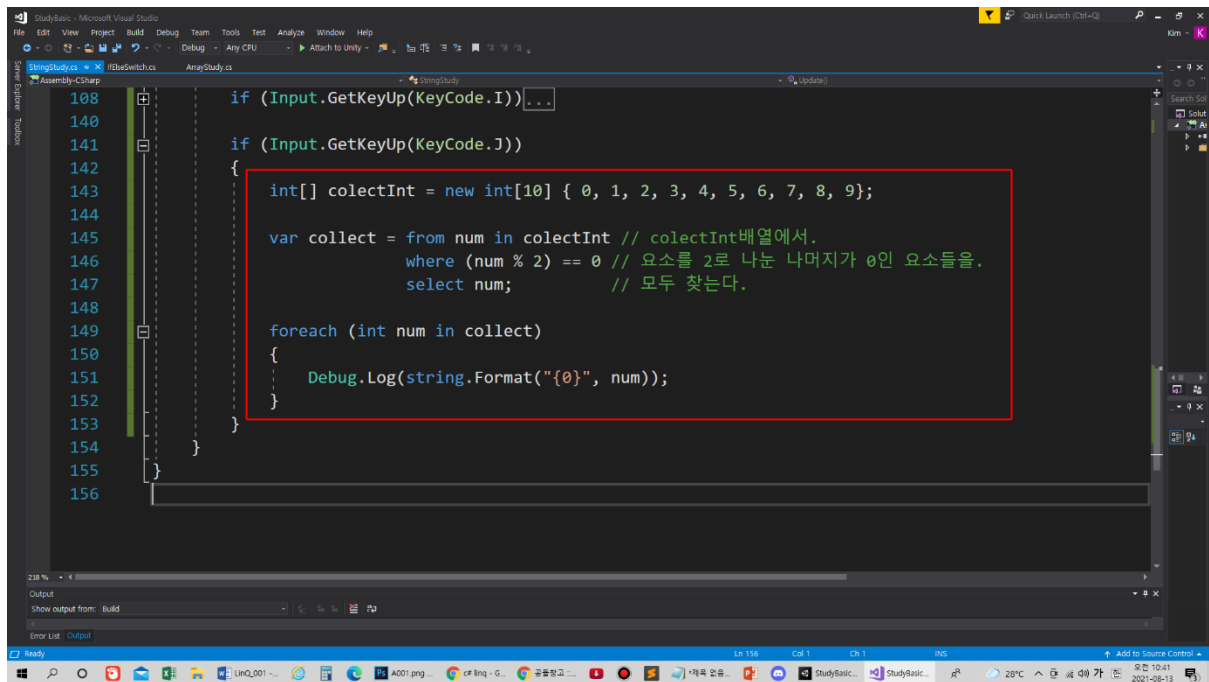
from : 어떤 데이터에서 찾을 것인가

where : 어떤 조건으로 찾을 것인가

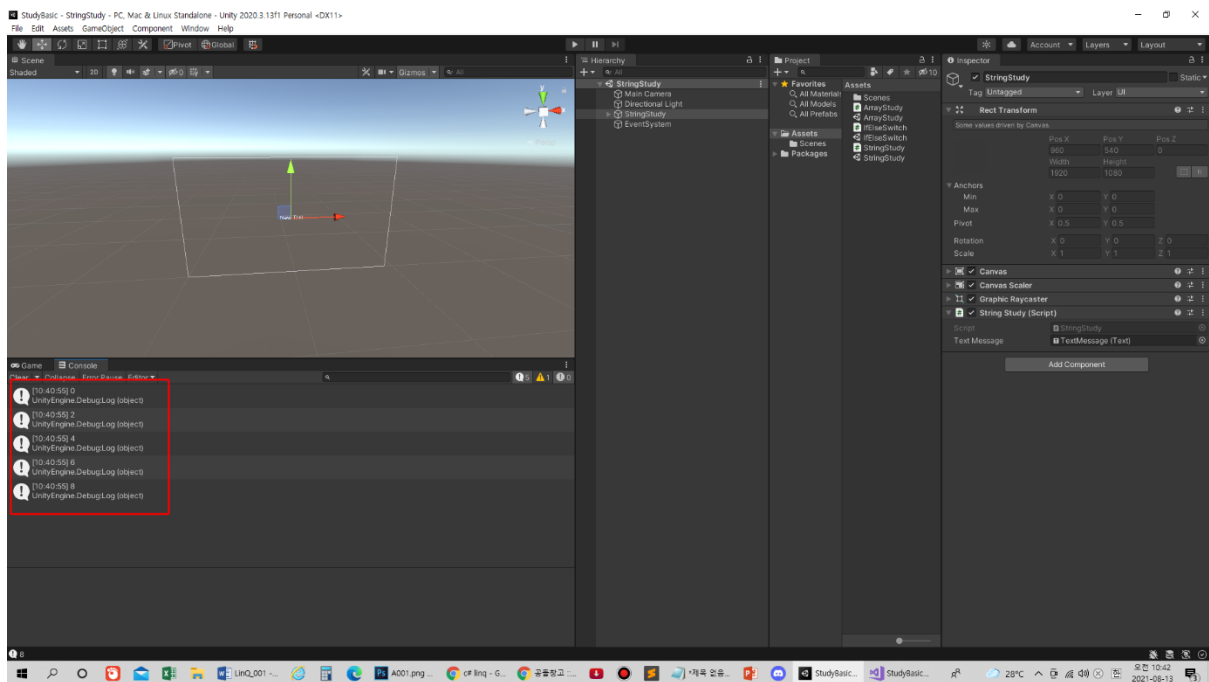
order by : 어떤 항목을 기준으로 정렬할 것인가

select : 어떤 항목을 추출할 것인가

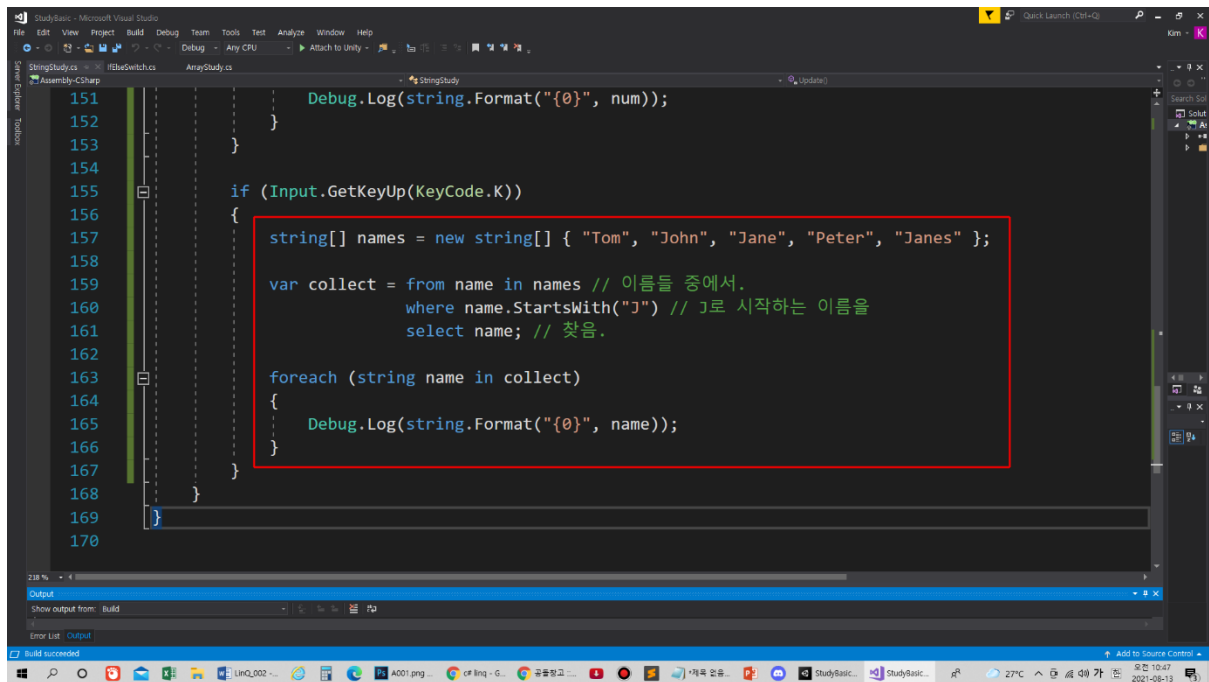
문장을 추가해 주도록 합니다.



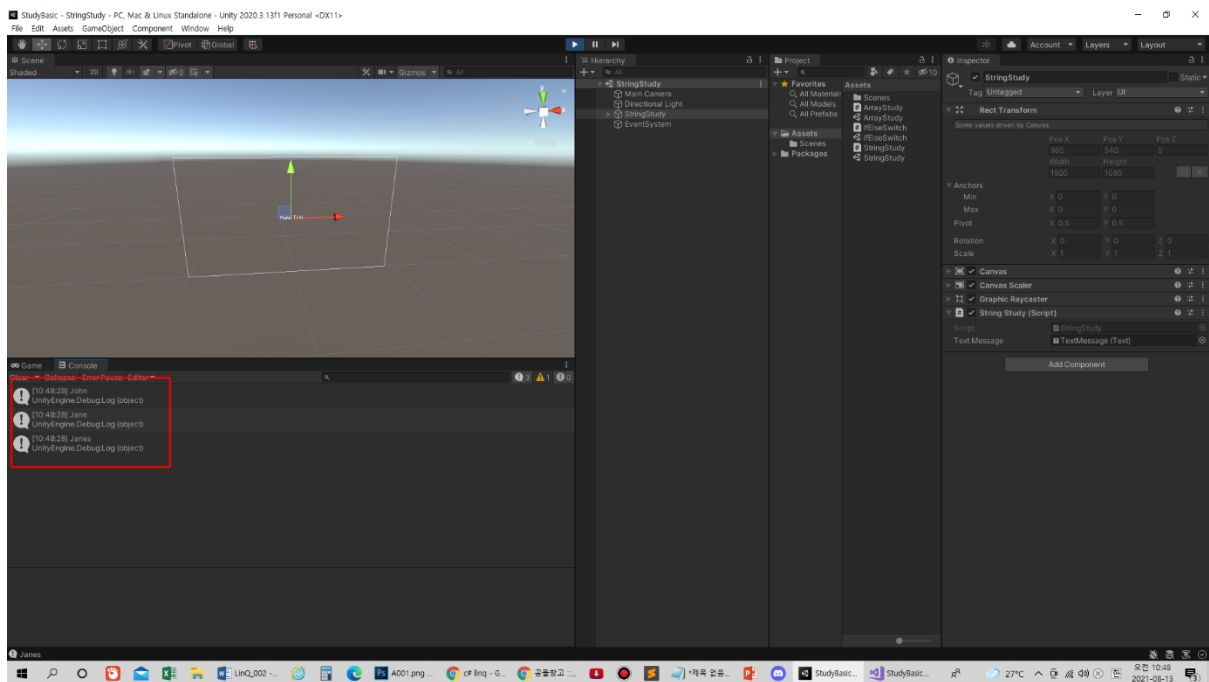
플레이를 하고 결과를 확인해 봅니다.

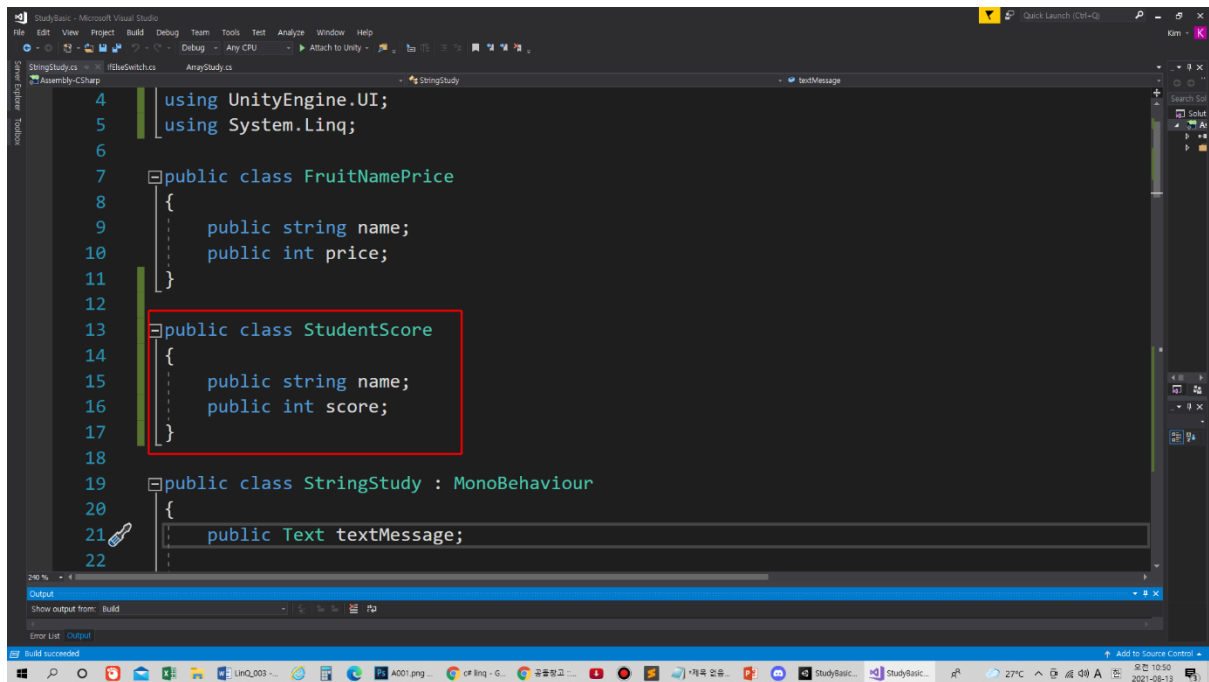


다음 문장을 보도록 합니다.

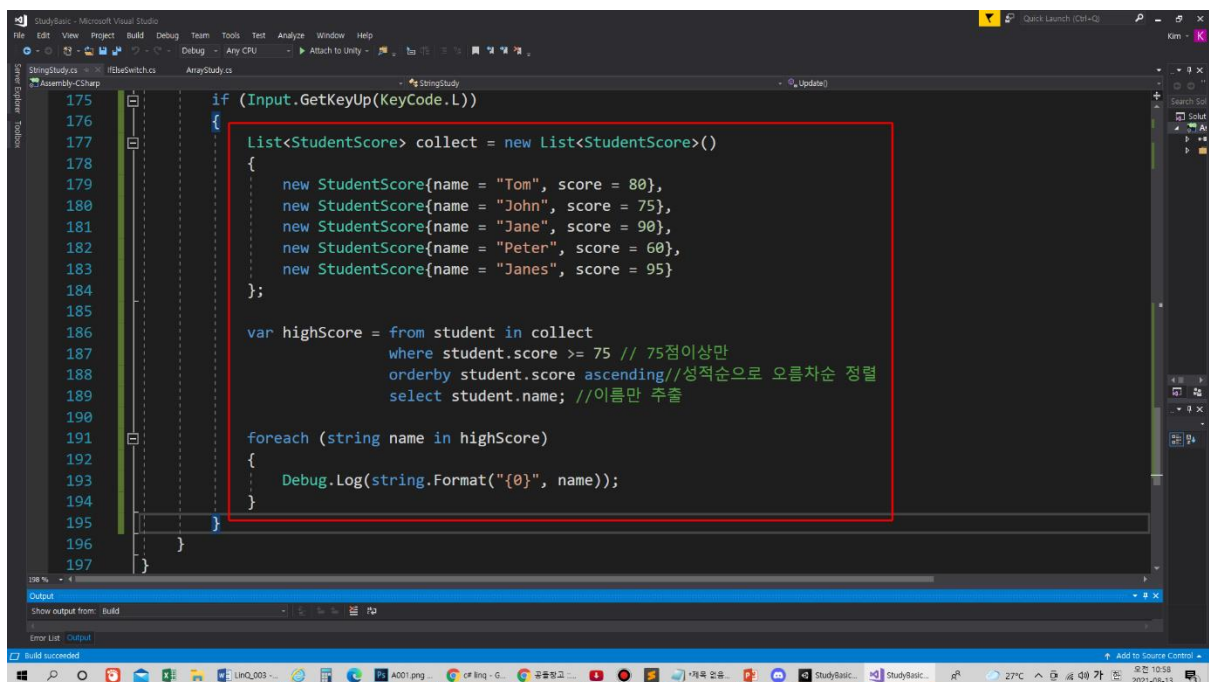


플레이를 하고 결과를 확인합니다.

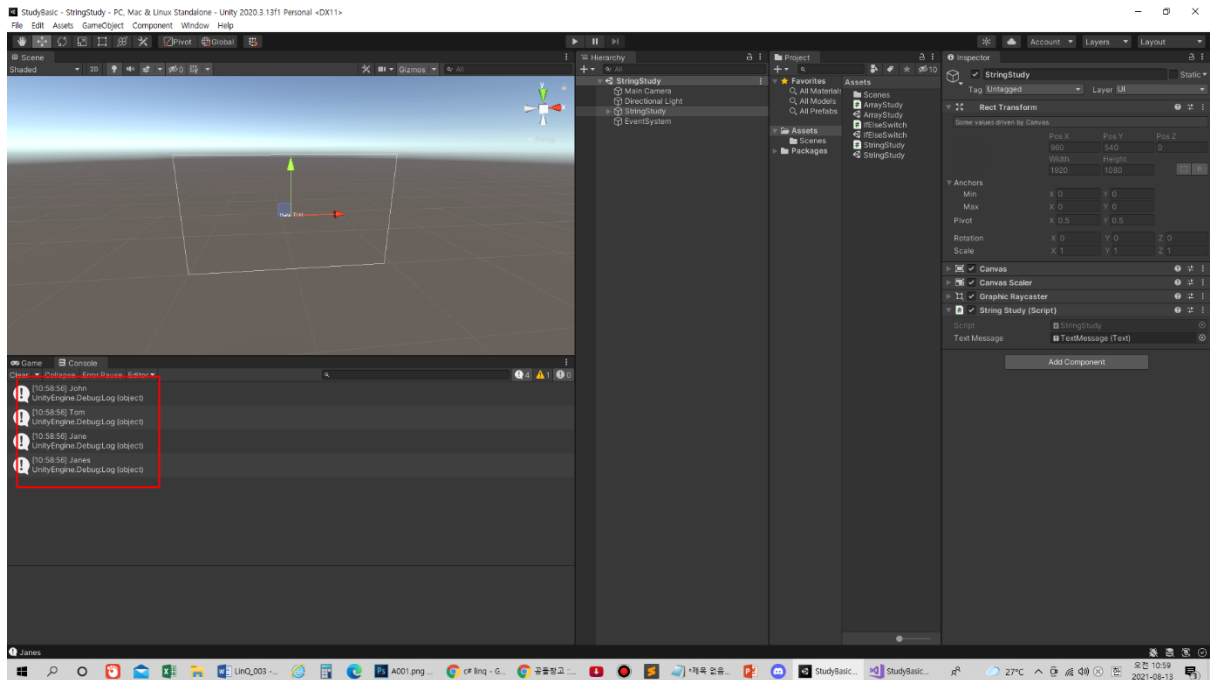




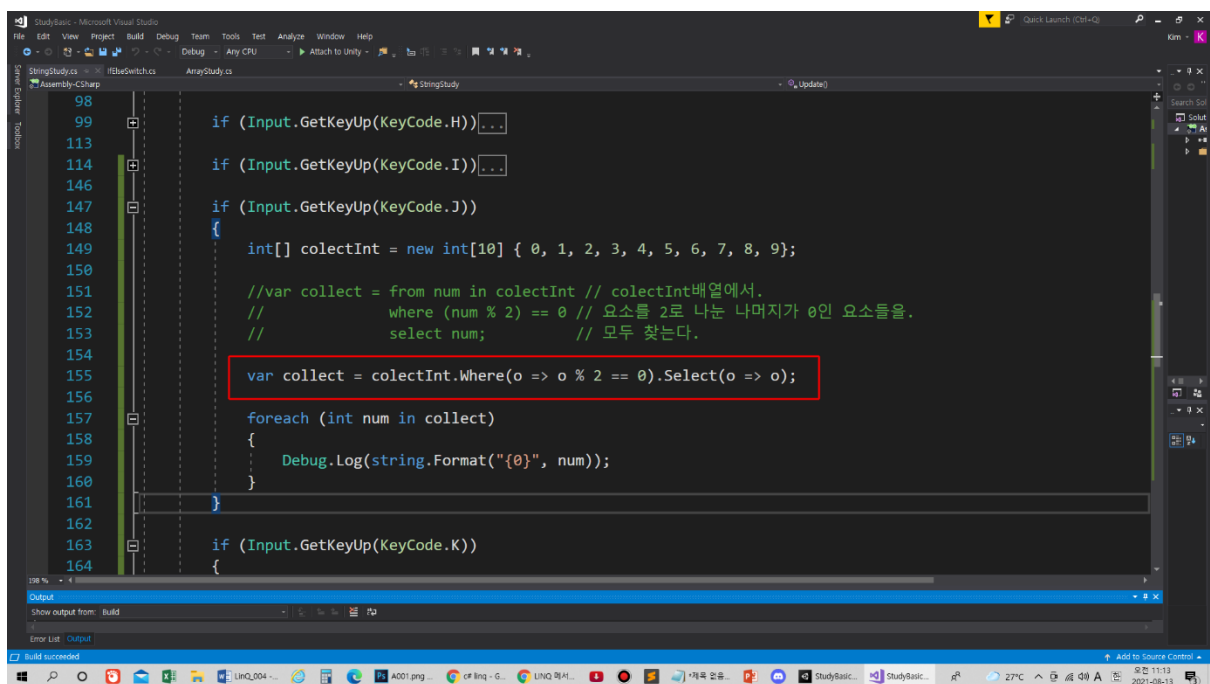
문장을 추가해 주도록 합니다.



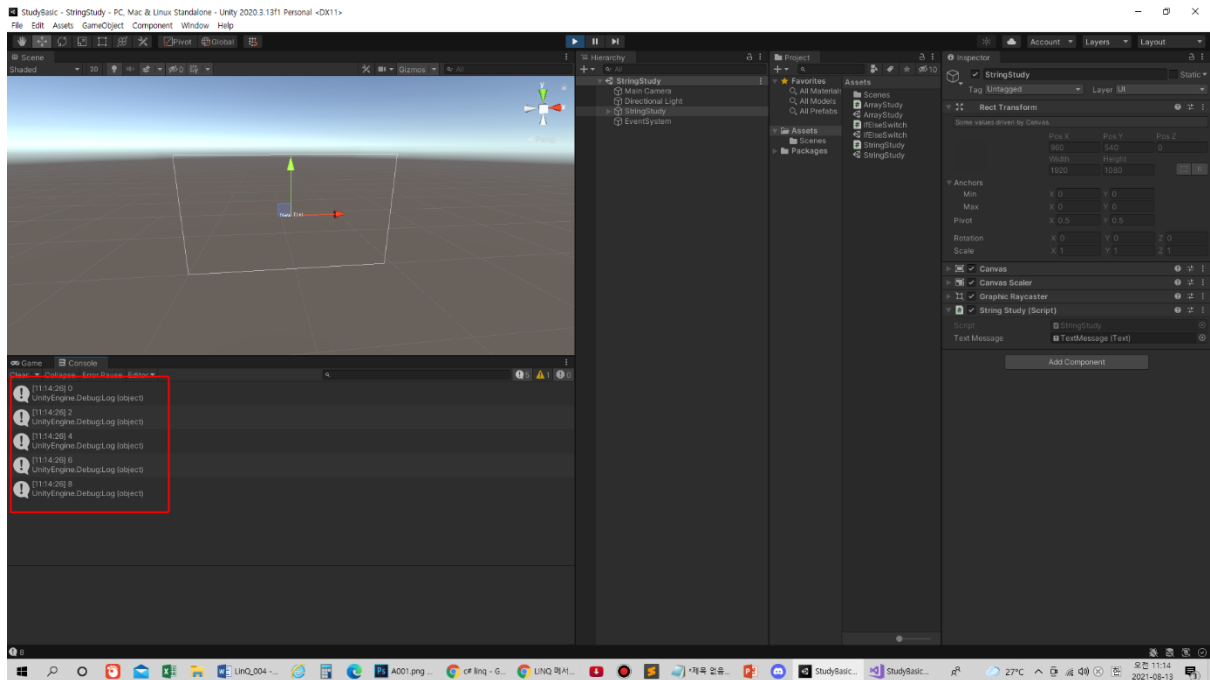
플레이를 해서 결과를 확인해 보도록 합니다.



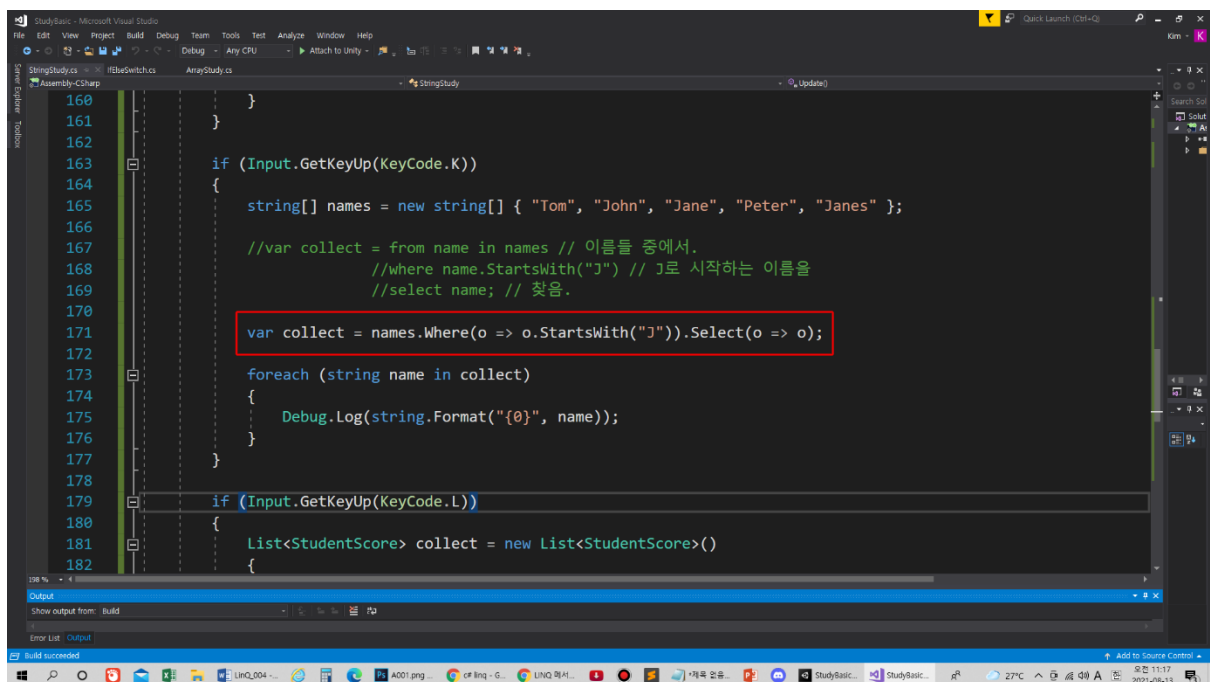
지금까지 쿼리식 표현을 알아 보았는데요. 메소드식 표현에 대해서 알아 보도록 합니다. 문장을 추가해 줍니다.



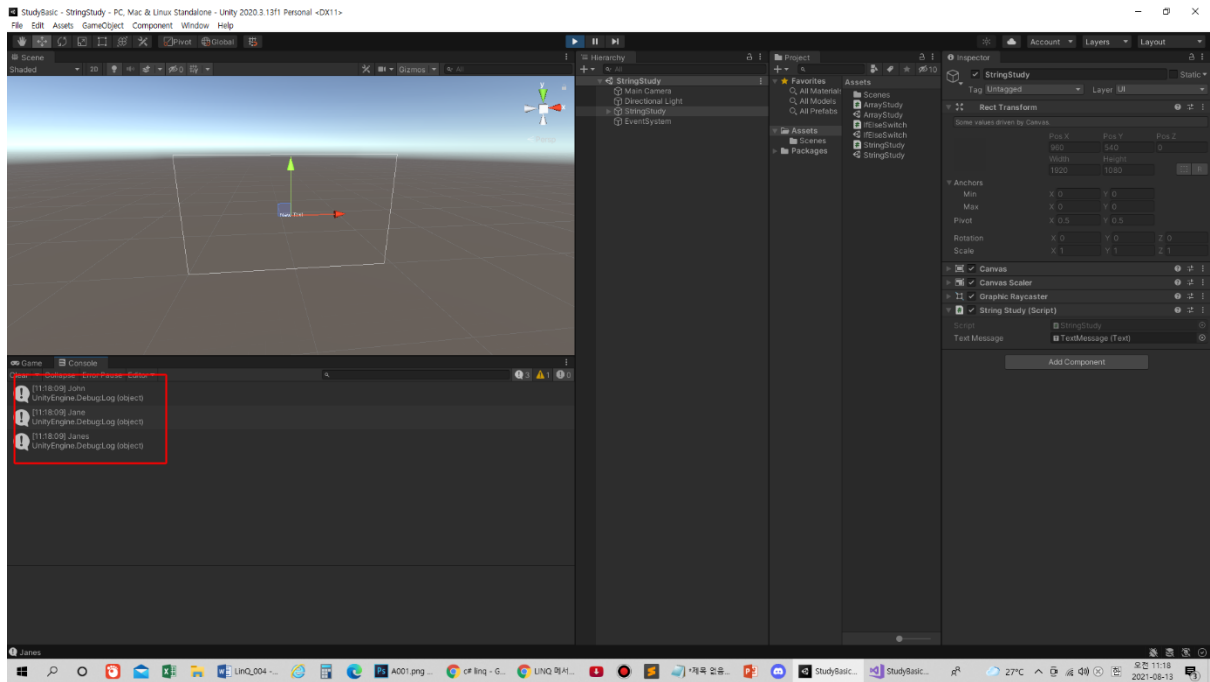
플레이를 하고 결과를 확인해 봅니다.



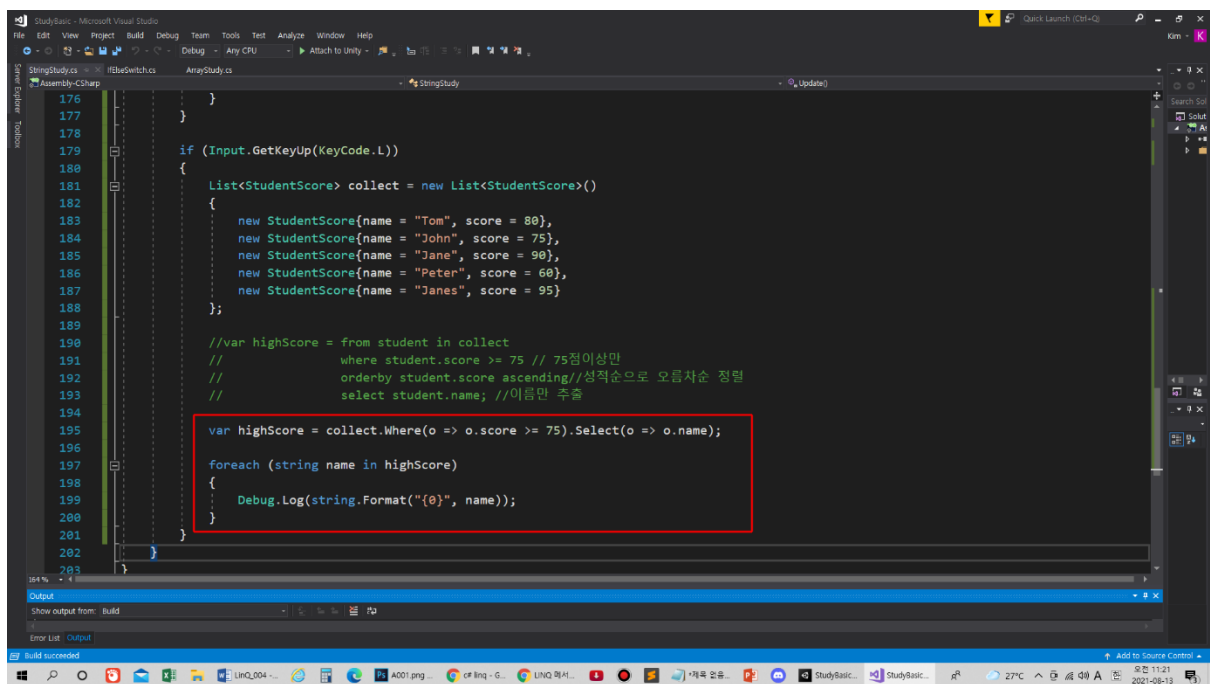
다음 문장을 수정해 주도록 합니다.



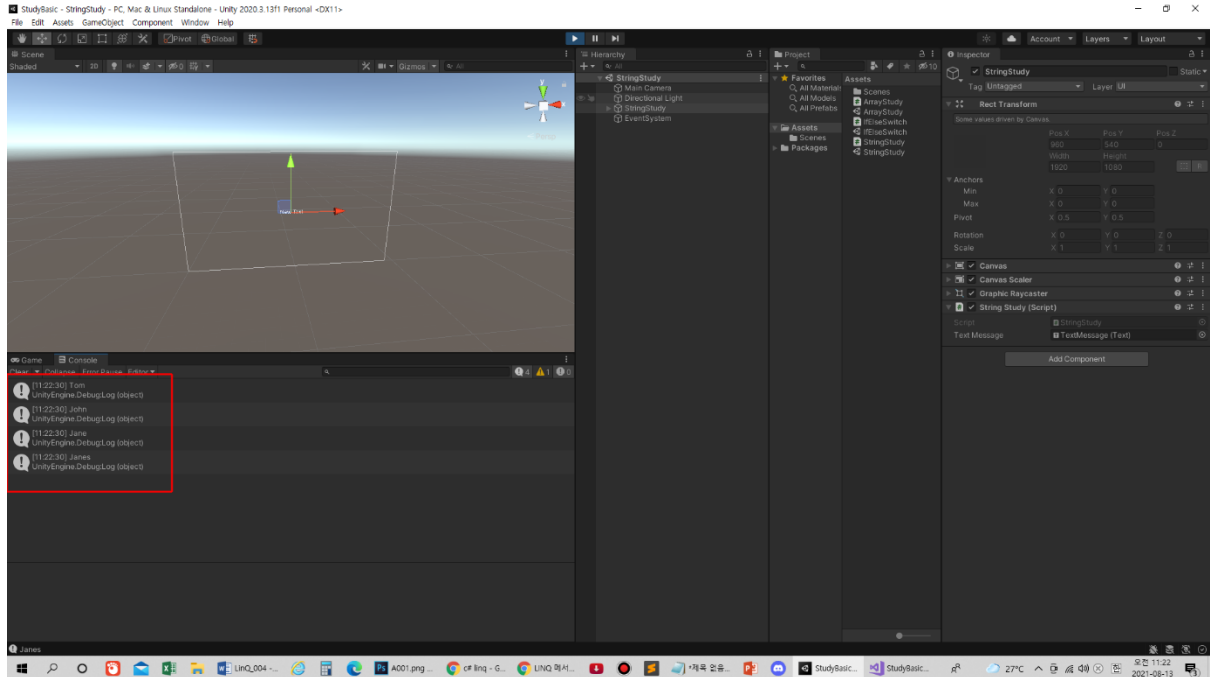
플레이를 하고 결과를 확인해 봅니다.



다음 문장을 수정해 주도록 합니다.



플레이를 하고 결과를 확인해 봅니다.



String 은 변경할 수 없는 형식입니다.

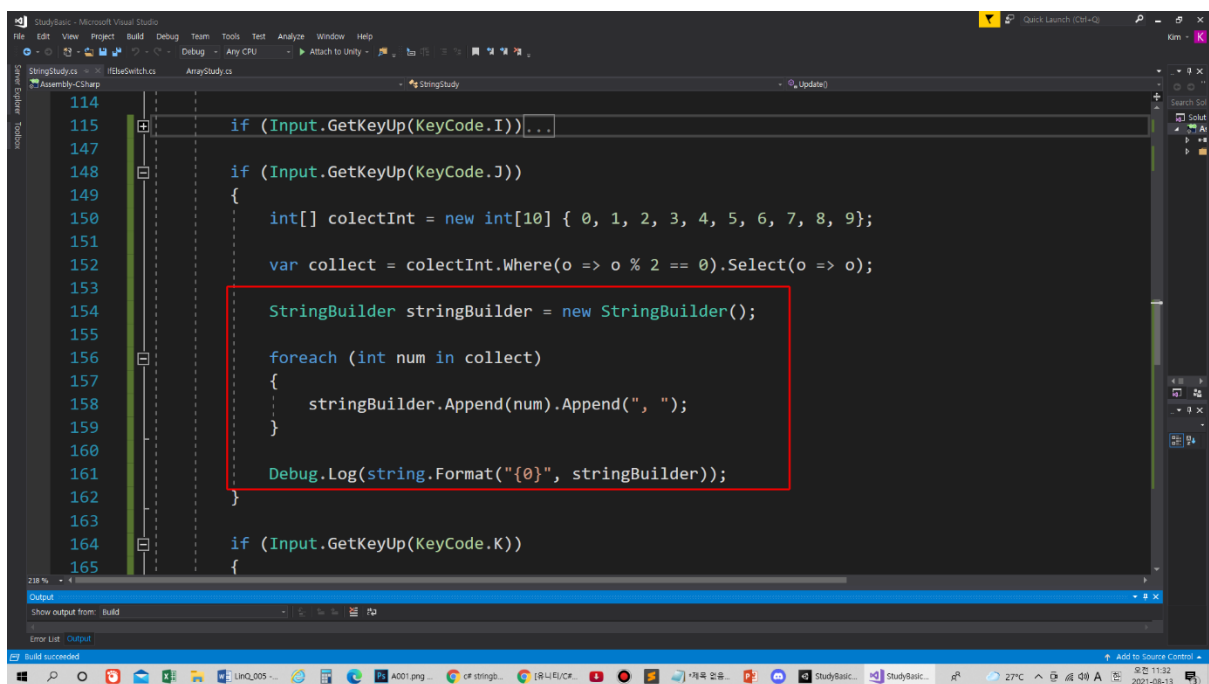
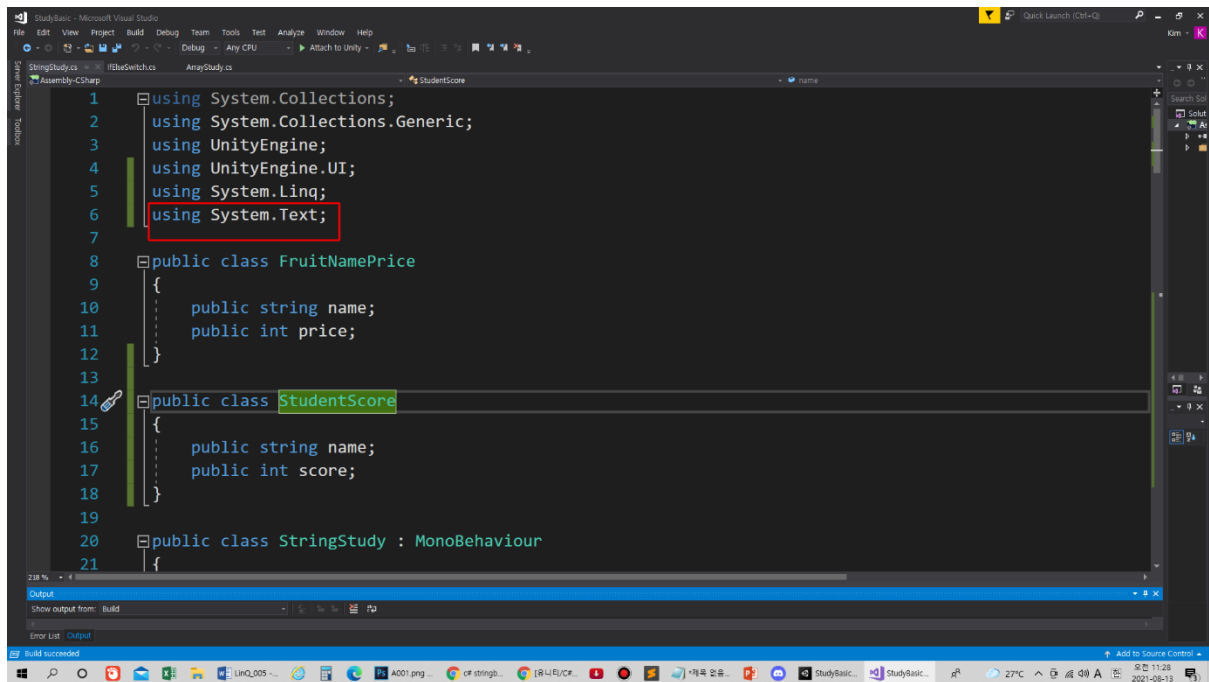
String 의 문자열을 변경을 하면 변경된 문자열을 위한 새로운 String 을 만들어 냅니다.

그래서 반복문같은 곳에서 계속 String 을 변경하면 성능 저하가 일어날 수 있습니다.

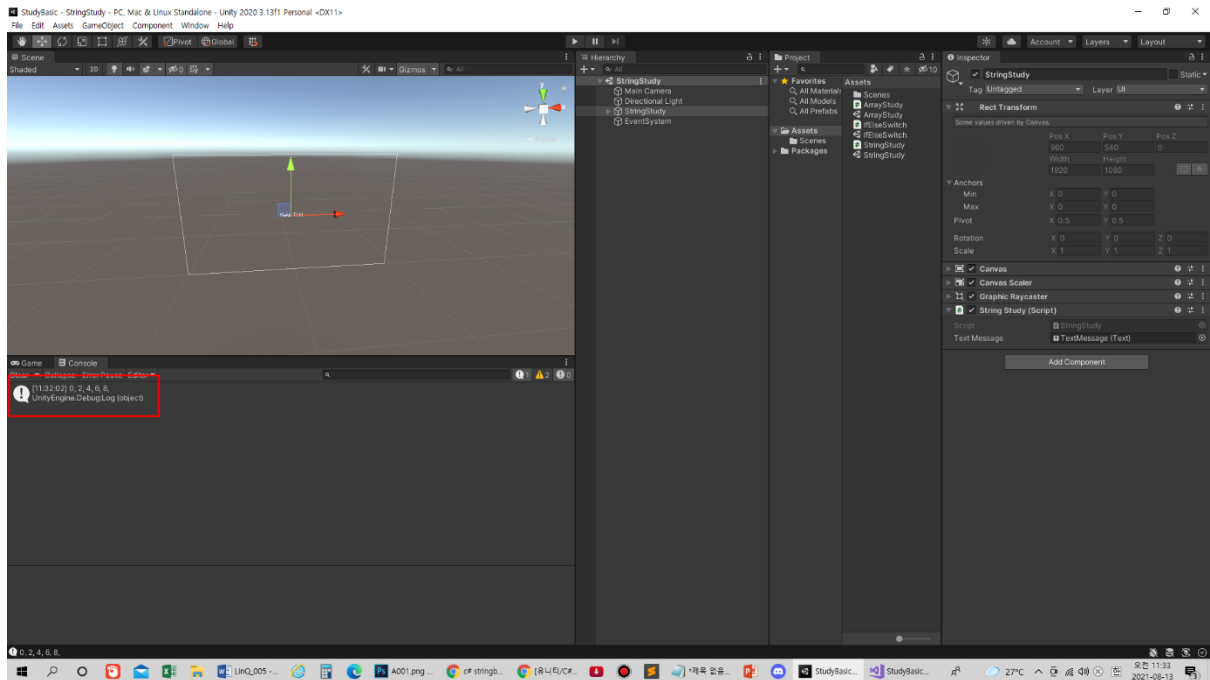
StringBuilder 는 변경 가능한 문자열 Class 입니다.

StringBuilder 를 사용하면 해결됩니다.

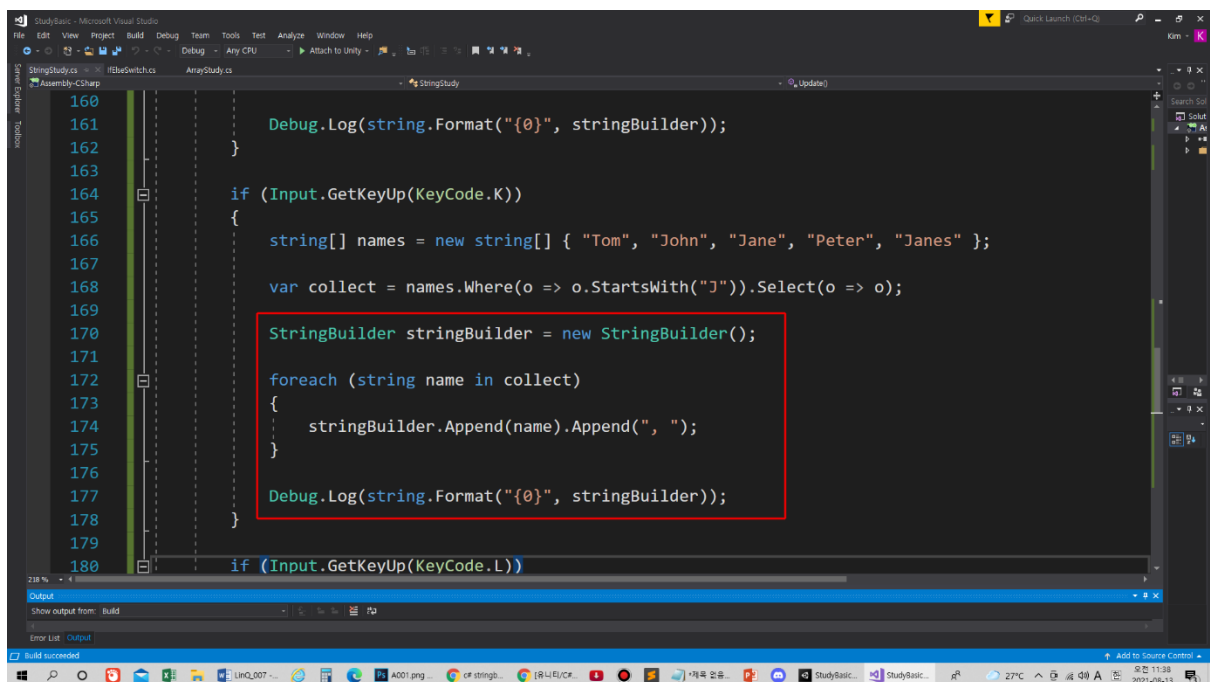
stringBuilder를 사용해 봅니다. 문장을 수정해 줍니다.



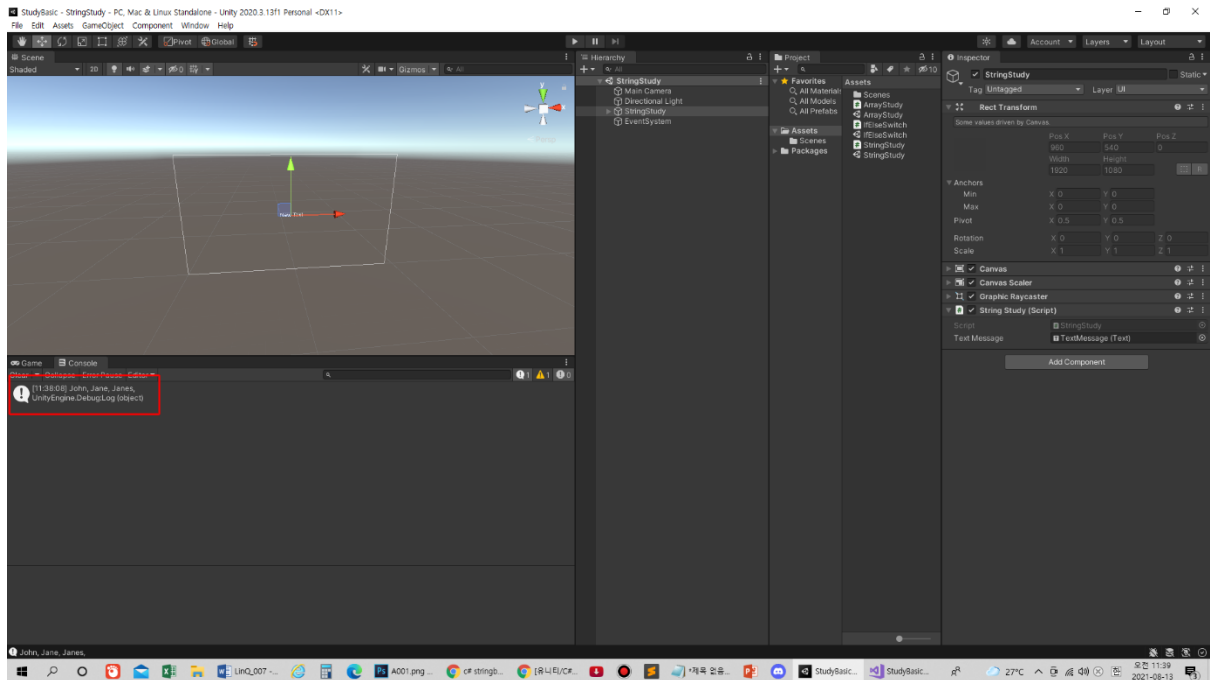
플레이를 해서 결과를 확인해 봅니다.



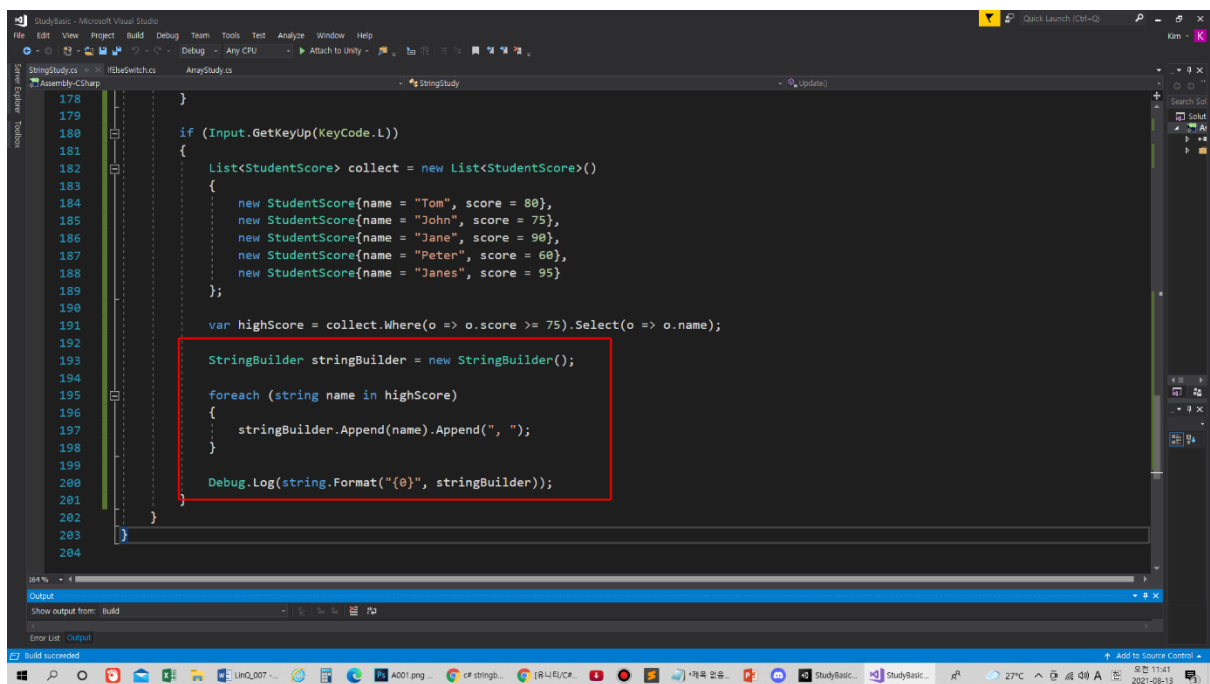
문장을 수정해 줍니다.



플레이를 하고 결과를 확인해 봅니다.



다음 문장을 수정해 주도록 합니다.



플레이를 하고 결과를 확인해 봅니다.

