[UTF-8 VS. UTF-8 BOM]

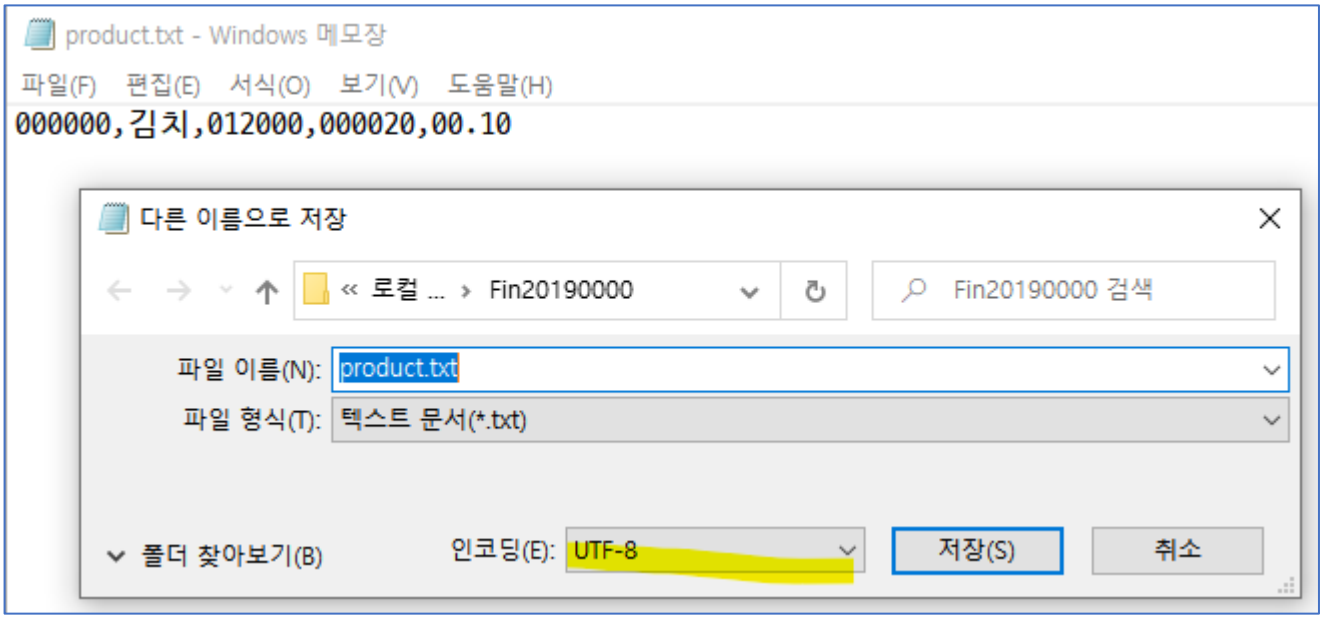
처음 상품 입력 시 파일에 생성할 시, 아래와 같은 코드로 작성을 하였습니다.

따로 인코딩 방식을 지정하지 않았습니다. (현재 시스템 상 UTF-8이 적용됩니다.)

```
var filename = Path.Combine(path, fileProduct);

using (var sw = new StreamWriter(new FileStream(filename, FileMode.Append)))
{
    sw.WriteLine($"{prod.Number.ToString("000000")},{prod.Name},{prod.UnitPri
}
```

생성한 파일을 메모장으로 열어서, '다른 이름으로 저장'하기를 하면 인코딩이 UTF-8임을 알 수 있습니다.

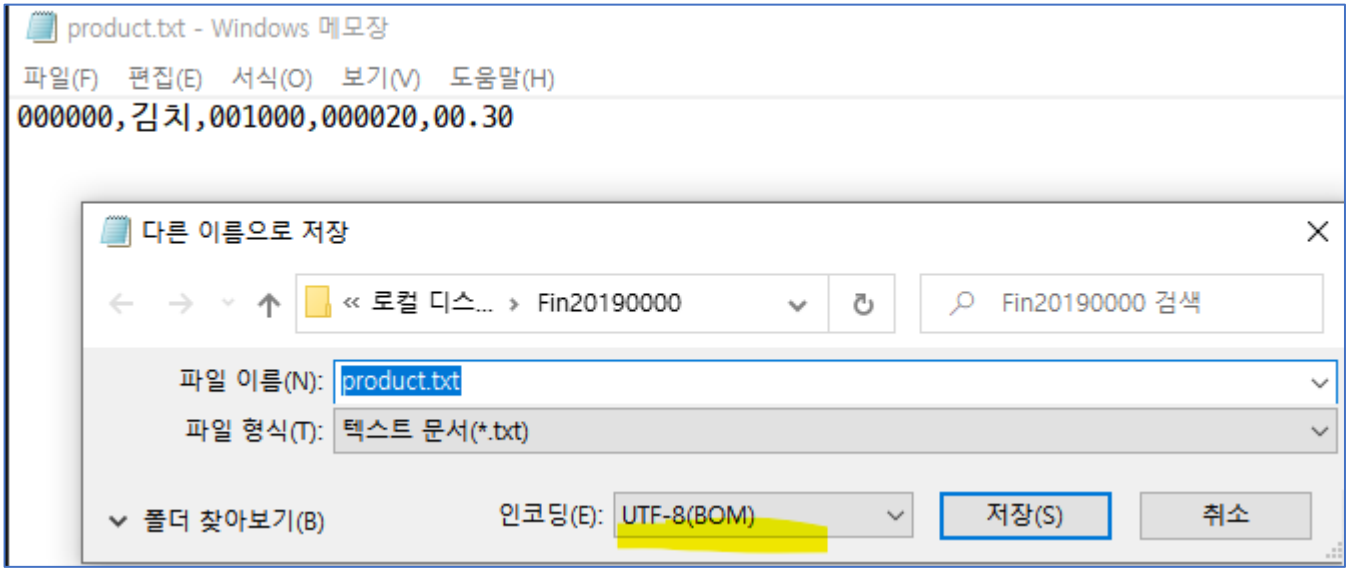


수업 도중에 인코딩 방식 사용하는 법을 추가하면서, 아래처럼 모든 코드에 UTF-8을 넣도록, 코드를 변경해보라고 했습니다.

```
var filename = Path.Combine(path, fileProduct);

using (var sw = new StreamWriter(new FileStream(filename, FileMode.Append), Encoding.UTF8))
{
    sw.WriteLine($"{prod.Number.ToString("000000")},{prod.Name},{prod.UnitPrice.ToString("00000
}
```

위 코드로 생성한 파일을 메모장에서 열어서, '다른 이름으로 저장'하기를 하면 인코딩이 UTF-8 (BOM) 임을 알 수 있습니다.



BOM (Byte Order Mark : EF BB BF) 라는 3바이트에 해당하는 데이터가 StreamWriter 사용시 UTF-8을 지정하면 자동으로 붙으면서 크기 계산에 문제가 생기고 있습니다.

이를 해결할 수 있는 방법은 두가지 입니다.

1) StreamWriter 사용시 Encoding.UTF8 인코딩 방법을 지정하지 않는다. (자동으로 UTF-8이 되지만 BOM이 붙지 않습니다.

2) StreamWriter 객체 생성시 BOM을 사용하지 않는 UTF-8 형식의 인코딩 방법을 사용합니다

```
var filename = Path.Combine(path, fileProduct);

using (var sw = new StreamWriter(new FileStream(filename, FileMode.Append), new UTF8Encoding(false)))
{
    sw.WriteLine($"{prod.Number.ToString("000000")},{prod.Name},{prod.UnitPrice.ToString("000000")},{ pro
}
```