

일반적으로 인스턴스를 생성할 때는 이렇게 생성자를 통해서 만들죠?

```
char data[] = {'a', 'b', 'c'};
String myString = new String(data);
```

```
abc
```

하지만 문자열의 경우에는 '문자열 리터럴(String Literal)'로 만들기 때문에 기본형이 아닌가 오해를 할 수도 있습니다.

```
String myString = "abc";
```

어쨌든 문자열도 클래스 기반의 자료형이기 때문에 여러 가지 메소드를 사용할 수 있는데요. 유용해보이는 몇 가지만 살펴보겠습니다.

대소문자 변환

toUpperCase 메소드를 이용하면 모든 글자를 대문자로, **toLowerCase** 메소드를 이용하면 모두 소문자로 변환할 수 있습니다.

```
String myString = "aBc";
System.out.println(myString.toUpperCase()) // 모두 대문자로
System.out.println(myString.toLowerCase()) // 모두 소문자로
System.out.println(myString)              // 그대로
```

```
ABC
abc
aBc
```

참고로 이 두 메소드는 **myString** 자체를 바꾸는 것이 아니라, 새로운 문자열을 리턴시켜주는 것입니다. 따라서 마지막에 **System.out.println(myString)** 을 하면 원래대로 **"aBc"** 가 나오는 거죠.

문자열 비교

```
String myString = "aBc";
System.out.println(myString.toLowerCase() == "abc");
```

```
false
```

양쪽 다 **"abc"** 이니까 **true** 가 나올 것 같은데, 실행하면 **false** 가 나오죠?

기본형의 비교 연산자는 양쪽의 값이 같으면 `true` 가 나오는데요. **참조형의 비교 연산자는 가리키는 인스턴스가 같은 인스턴스인지 확인하는 역할을 합니다.** 위의 예시에서 두 인스턴스가 서로 다른 인스턴스이기 때문에 `false` 가 나오는 거죠.

만약 두 문자열의 **내용**이 같은지 비교하려면 `==` 가 아니라 `equals` 메소드를 사용하셔야 합니다.

```
String myString = "aBc";
System.out.println(myString.toLowerCase().equals("abc"));
```

```
true
```

예전에 'DNA 염기 서열' 문제에서는 `equals` 를 쓰지 않고 `switch` 문에 넣었는데, 이건 왜 제대로 작동했을까요? `switch` 문 내부에서 `==` 대신 `equals` 메소드를 쓰고 있었기 때문입니다!



수업을 완료하셨으면 체크해주세요.



수강생 Q&A 보기



(/questions? 질문하기

assignment_id=354&sort_by=popular)
(/questions/new?

assignment_id=354&op1=%EA%B0%9D%EC%B2%B4+%EC%A7

< 이전 강의
단위 변환기 (해설) (/assignments/445)

다음 강의 > (/assignments/355)
숫자 도구