

'Wrapper 클래스'는 기본 자료형을 객체 형식으로 감싸는 역할을 합니다. **Integer** 클래스는 **int** 형을, **Double** 클래스는 **double** 을, **Long** 클래스는 **long** 을, **Boolean** 클래스는 **boolean** 을 감싸는 것이죠.

그런데 이런 Wrapper 클래스가 왜 필요할까요?

기본형 자료형(Primitive Type)을 참조형(Reference Type)처럼 다루어야 할 때 Wrapper 클래스를 사용하면 됩니다! 예를 들어서 바로 다음 강의에서 사용하는 **ArrayList** 같은 컬렉션을 사용할 때 꼭 참조형을 사용해야 합니다.

Wrapper 클래스의 인스턴스는 생성자로 생성할 수도 있고, 리터럴로 생성할 수도 있습니다.

```
Integer i = new Integer(123);
Integer i = 123;
```

그런데 주의할 점이 있습니다.

```
System.out.println(123 == 123);
System.out.println(new Integer(123) == new Integer(123));
```

```
true
false
```

첫 번째 줄의 결과값은 예상대로 **true** 입니다. 그런데 왜 두 번째 줄은 **false** 가 나올까요? 참조형의 비교 연산자는 가리키는 객체가 같을지 비교하기 때문입니다. 두 생성자를 통해 만든 다른 객체이기 때문에 **false** 가 나오는 거죠.

두 **String** 인스턴스를 비교할 때처럼 **equals** 메소드를 쓰면 '값'이 같은지 확인할 수 있습니다.

```
System.out.println(new Integer(123).equals(new Integer(123)));
```

```
true
```



수업을 완료하셨으면 체크해주세요.



수강생 Q&A 보기



질문하기

assignment\_id=372&sort\_by=popular)  
(/questions/new?

assignment\_id=372&op1=%EA%B0%9D%EC%B2%B4+%EC%A7

< 이전 강의  
숫자 도구 (/assignments/355)

다음 강의 > (/assignments/371)  
ArrayList