

Wrapper Classes in Java

First, the Problem

- **Primitive types** → int, float, double, char, boolean etc. (simple, fast, not objects).
- **Objects** → Every class we create (String, ArrayList, etc.) are objects.

But **Collections (like ArrayList, HashMap)** can only store **objects**, not primitives.

So, what if we want to put an int inside an ArrayList?

Wrapper Class

- A **Wrapper Class** is a class in Java that “wraps” a primitive type into an object form.
- Each primitive has a corresponding wrapper class:

Primitive Type	Wrapper Class
byte	Byte
short	Short
int	Integer
long	Long
float	Float
double	Double
char	Character
boolean	Boolean

Why Wrapper Classes?

- **To use primitives in Collections** (e.g., `ArrayList<Integer>` instead of `ArrayList<int>`).
- **They provide useful methods** (e.g., converting string "123" to integer using `Integer.parseInt("123")`).
- Needed for **generics**, because generics only work with objects.

Autoboxing and Unboxing

- Earlier you had to manually convert between primitives and wrappers:
 - `int x = 10;`
 - `Integer y = Integer.valueOf(x); // boxing`
 - `int z = y.intValue(); // unboxing`
- From Java 5 onwards: **Autoboxing** and **Unboxing** automatically do this.
 - `int x = 10;`
 - `Integer y = x; // autoboxing`
 - `int z = y; // unboxing`

Quick Examples

- Example 1: Using wrapper in collections

```
ArrayList<Integer> list = new ArrayList<>();  
list.add(10); // primitive int converted to Integer automatically  
list.add(20);
```

```
System.out.println(list); // [10, 20]
```

Example 2: Converting string to number

```
String s = "123";
```

```
int num = Integer.parseInt(s); // convert String → int
```

```
System.out.println(num + 1); // 124
```


One-Line Definition

- Wrapper classes in Java are special classes that convert primitive data types into objects, so they can be used in collections, generics, and provide utility methods.

Immutability:

- Wrapper classes are **immutable** (like String).
- Once created, their value cannot change.

- Example

```
Integer a = 10;
```

```
a = a + 5; // creates a new Integer object with value 15
```