



# Wrapper 클래스

자바는 기본타입의 값을 갖는 객체를 생성할 수 있다.  
이런 객체를 포장객체 라고 한다. (기본타입의 값을 내부에 갖고있는 객체)

포장 객체의 설계도인 포장클래스는 java.lang 패키지에 포함되어 있는데, 기본타입에 대응되는 클래스들이 있다.

기본타입	포장클래스
byte	Byte
char	Character
short	Short
int	Integer
long	Long
float	Float
double	Double
boolean	Boolean



# 박싱(Boxing)

기본타입의 값을 포장객체로 만드는 과정을 박싱이라고 한다.

박싱(Boxing)
<code>Byte obj = new Byte(10);</code>
<code>Character obj = new Character('가');</code>
<code>Short obj = new Short(100);</code>
<code>Integer obj = new Integer(1000);</code>
<code>Long obj = new Long(10000);</code>
<code>Float obj = new Float(2.5F);</code>
<code>Double obj = new Double(3.5);</code>
<code>Boolean obj = new Boolean(true);</code>



# 언박싱(Unboxing)

박싱된 포장객체에서 다시 기본타입의 값을 얻어내기 위해서는 각 포장클래스마다 가지고 있는 '기본타입이름 + Value()'메소드를 호출하면 된다.

언박싱(Unboxing)
<code>byte num = obj.byteValue();</code>
<code>char ch = obj.charValue();</code>
<code>short num = obj.shortValue();</code>
<code>int num = obj.intValue();</code>
<code>long num = obj.longValue();</code>
<code>float num = obj.floatValue();</code>
<code>double num = obj.doubleValue();</code>
<code>Boolean bool = obj.booleanValue();</code>



# 자동박싱과 언박싱

기본타입의 값을 직접 박싱하지 않아도 자동박싱이 일어난다.

```
Integer obj = 100; [자동박싱]
```

wrapper 클래스 객체에서 직접 언박싱하지 않아도 자동언박싱이 일어난다.

```
Integer obj = new Integer(200);
```

```
int value = obj; [자동언박싱]
```