

为了能够将这些基本数据类型当成对象操作，Java为每一个基本数据类型都引入了对应的包装类型（wrapper class）。**基本类型的包装类都是不可变的（都是final修饰的）**。从Java 5开始引入了自动装箱/拆箱机制。

原始类型：boolean, char, byte, short, int, long, float, double

包装类型：Boolean, Character, Byte, Short, Integer, Long,

Float, Double

自动装箱：valueOf（）

自动拆箱：intValue, floatValue等等（Integer遇int会自动拆箱为int）

包装类的parseXXX方法还可以实现不同类型之间的转换, 如数字字符串转换为int类型：
`int i = Integer.parseInt("23");`

例子：

```
1. public class TestInteger {
2.     public static void main(String[] args) {
3.         int i = 128;
4.         Integer i2 = 128;                                //编译时
被翻译成: Integer i4 = Integer.valueOf(128);从而使用常量
池中的对象。对于五种包装类：-128到127之间的数，会进行缓
存,应该字面量在常量池。超出范围则在堆中创建对象，应该相当
Integer x = new Integer(128)
5.         Integer i3 = new Integer(128);                //对象在
堆中
```

6. `System.out.println(i == i2);` //Integer遇int会自动拆箱为int，所以为true

```
Integer i4 = new Integer(40);
Integer i5 = new Integer(40);
Integer i6 = new Integer(0);

System.out.println("i4=i5+i6    " + (i4 == i5 + i6));
```

结果：

i4=i5+i6 true

解释：

语句 `i4 == i5 + i6`，因为+这个操作符不适用于 Integer 对象，首先 i5 和 i6 进行自动拆箱操作，进行数值相加，即 `i4 == 40`。