



第3章 变量与数据类型

3.1 为什么需要变量？

- **程序即世界**：程序如一个虚拟世界，变量就像世界中的**储物格**——快递柜中的每个格子（变量）存放不同的包裹（数据）。
 -  **现实场景**：
 - 收快递时，包裹放在A12号柜子（变量名 `a12`），柜子里的物品是书本（变量值 `"Java教程"`）。
 -  **核心意义**：
 - **灵活存储数据**：可随时修改柜子里的内容（变量值变化）。
 - **代码可读性**：变量名如 `age` 代替纯数字 `30`，代码含义一目了然。

3.2 变量是什么？怎么用？

1. **变量的本质**：内存中的一块存储空间，**通过变量名访问数据**。

- **门牌号比喻**：

变量名 → 门牌号（如 `Room101`）
变量值 → 房间内的物品（如 `Java课本`）

2. **使用三步走**：

1. 声明变量：`int age;` *// 类型 + 变量名*
2. 赋值：`age = 18;` *// 将18放进age变量*
3. 使用：`System.out.println(age);` *// 输出18*

- **可一步完成**：`int money = 100;`

3.3 变量使用入门（代码实战）

案例：用户个人信息存储

```
public class UserInfo {
    public static void main(String[] args) {
        String name = "李雷";    // 字符串类型变量
        int age = 25;            // 整数类型变量
        double height = 1.75;    // 浮点数类型变量
        char gender = '男';      // 字符类型（单引号包裹）

        System.out.println(name);    // 输出：李雷
        System.out.println(age + "岁"); // 输出：25岁
        System.out.println("身高：" + height + "米");

    }
}
```

! 3.4 变量使用注意事项

1. 先声明，后使用：

```
int count;           // ✓ 先声明
count = 10;          // ✓ 再赋值
System.out.println(b); // ✗ 错误！变量b未声明
```

2. 同一作用域内不重名：

```
int score = 90;
int score = 85;      // ✗ 错误！重复声明score
```

3. 变量值可变，但类型不可变：

```
int level = 1;
level = 2;           // ✓ 允许修改值
level = "VIP";        // ✗ 错误！类型不可变
```

4. 变量未赋值时？直接使用会报错！

🌈 3.5 Java数据类型体系（完整分类）

类型	关键字	大小	取值范围	示例
整数类型	byte	1字节	-128 ~ 127	byte a = 100;
	short	2字节	-32768 ~ 32767	short b = 20000;
	int	4字节	-21亿 ~ 21亿	int c = 1000000;
	long	8字节	极大(需加L)	long d = 100000000000L;
浮点类型	float	4字节	小数(需加F)精度低	float e = 3.14F;
	double	8字节	小数，默认类型精度高	double f = 3.1415926535;
字符类型	char	2字节	单个字符/Unicode码	char g = 'A';
布尔类型	boolean	1字节	true / false	boolean h = true;

! 为什么选择 double 而非 float？

3.14（默认是double）→ float需强制转换：`float x = 3.14F;`

1234 3.6 基本数据类型转换（自动 vs 强制）

1. 自动类型转换（小 → 大）：

```
int a = 100;
double b = a; // ✓ int自动转为double → 100.0
```

◦ 规则: byte → short → int → long → float → double

2. 强制类型转换 (大 → 小, 可能丢失精度) :

```
double money = 88.8;
int cash = (int) money; // ✓ 强制转换 → cash=88 (丢失小数部分)
```

3. 特殊情况 (char与整数转换) :

```
char c = 'A';
int code = c; // 自动转换 → code=65
char d = (char) 66; // 强制转换 → d='B'
```

3.7 String与基本类型转换

1. 基本类型 → String:

```
int age = 20;
String s1 = age + ""; // ➡ "20"
String s2 = String.valueOf(3.14); // ➡ "3.14"
```

2. String → 基本类型:

```
String str = "123";
int num = Integer.parseInt(str); // ➡ 123
double pi = Double.parseDouble("3.14"); // ➡ 3.14
```

3. 注意事项:

- 格式必须合法: Integer.parseInt("abc") 会抛出异常!
- 字符串转字符: char c = "Hello".charAt(0); (得到第一个字符 H)

3.8 综合实战: 变量与数据类型应用

任务: 计算商品总价并输出

```
public class ProductPrice {
    public static void main(String[] args) {
        String productName = "Java编程书";
        int quantity = 3; // 购买数量
        double price = 59.9; // 单价
        double total = quantity * price;

        System.out.println("商品名: " + productName);
        System.out.println("总金额: " + total + "元");
    }
}
```

运行结果：

商品名：Java编程书
总金额：179.7元

本章总结

变量是程序和数据的桥梁，就像快递柜的格子——**命名规范**是门牌号，**数据类型**决定格子大

赋值操作是存取物品。通过本章，你已掌握了Java世界中“储物”的核心法则！📦🔑

★ 全套练习题（附答案） ★

一、选择题（每题2分）

1. 以下变量声明正确的是：

- A. `int 3a = 10;`
- B. `float price = 3.14;`
- C. `char gender = "男";`
- D. `double score = 99.5;`

2. `long` 类型变量赋值时，必须加的后缀是：

- A. `L`
- B. `F`
- C. `D`
- D. 无需后缀

3. 以下能正确存储中文字符的是：

- A. `char c = 'A';`
- B. `char c = '中';`
- C. `String s = "中";`
- D. `int n = '中';`

4. `double result = 1 / 2;` 的值是：

- A. 0.5
- B. 0
- C. 0.0
- D. 编译错误

5. 以下程序段输出结果是：

```
System.out.println('A' + 1);
```

- A. `A1`
- B. `66`
- C. `B`
- D. 编译错误

6. `int` 类型转换为 `double` 属于：
- A. 自动类型转换
 - B. 强制类型转换
 - C. 可能会丢失精度
 - D. 以上均否
7. 以下不会编译错误的是：
- A. `int a = "123";`
 - B. `String s = 100 + "";`
 - C. `double d = 3.14F;`
 - D. `boolean b = 1;`
8. 比较两个浮点数 `a` 和 `b` 的正确方法是：
- A. `a == b`
 - B. `a - b < 0.5`
 - C. `Math.abs(a - b) < 1e-6`
 - D. 直接判等
9. Java中默认的浮点类型是：
- A. `float`
 - B. `double`
 - C. `int`
 - D. `long`
10. `String` 转 `int` 的正确方法是：
- A. `(int)str`
 - B. `Integer.parseInt(str)`
 - C. `str.toInt()`
 - D. `Int.valueOf(str)`

二、填空题（每空2分）

1. Java的基本数据类型共有___种。
2. 声明 `float` 类型变量时，值需加后缀___。
3. `char` 类型本质上存储的是___值。
4. `System.out.println((int)'A');` 的输出是___。
5. 强制类型转换 `double` 到 `int` 的语法是： `int x = (_____) 3.14;`
6. `byte` 类型的取值范围是___到___。
7. 字符串 `"123"` 转 `double` 的代码是： `double d = _____.parseDouble("123");`
8. 自动类型转换的方向遵循___ → `long` → `float` → `double` 的优先级。
9. `boolean` 类型的可能取值是___和___。
10. `System.out.println(5 / 2);` 的输出是___。

三、判断题（每题1分）

1. `char` 类型可以存储中文字符。()
2. `float a = 3.14;` 是合法的。()
3. 变量的作用域内可以声明同名变量。()
4. `boolean` 类型可以参与算术运算。()
5. Java中 `long` 类型占用8字节。()

6. `String` 是基本数据类型。()
 7. `int x = 'A' + 1;` 的 `x` 值为 `B`。()
 8. `byte b = 128;` 会编译错误。()
 9. 类型转换时, 从 `double` 到 `float` 是自动完成的。()
 10. `System.out.println(0.1 + 0.2 == 0.3);` 的输出是 `true`。()
-

四、简答题 (每题5分)

1. 描述Java的8种基本数据类型及其适用场景。
 2. 解释自动类型转换和强制类型转换的区别。
 3. 为什么浮点数不建议直接判等? 如何正确比较?
 4. 什么是变量的作用域? 试举违反作用域规则的例子。
 5. 编写代码将字符串 `"3.14"` 转为 `double` 类型。
-

五、编程题 (每题10分)

1. **变量声明与计算**: 声明两个整数变量 `a=10`、`b=20`, 计算它们的和并输出。
2. **类型转换**: 定义一个 `double` 变量 `d = 3.14159`, 将其强制转换为 `int` 输出。
3. **用户信息存储**: 声明变量存储用户信息 (姓名、年龄、身高、性别), 并打印输出。
4. **浮点数精度处理**: 计算 `0.1 + 0.2`, 判断是否等于 `0.3` (使用正确方法)。
5. **String转换**: 编写代码将整数 `100` 转为字符串, 再将 `"3.14"` 转为 `double` 类型。