



数据在计算机中的存储

所有数据最终都以二进制（0 和 1）形式存储在计算机的内存中
存储单位是字节（Byte）
不同类型的数据有不同的编码方式和内存布局

- 1 bit = 一个二进制位（0 或 1）
- 1 byte = 8 bits

现代计算机中，内存按 **字节地址寻址**，即：

地址0x0000 存储第1个字节
地址0x0001 存储第2个字节
...

1234 基本数据类型

数据类型	大小	存储内容	举例（值）	二进制表示（补码）
int	4 byte	整数	5	00000000 00000000 00000000 00000101
float	4 byte	浮点数	3.14f	IEEE754编码方式
char	2 byte	Unicode字符	'A'	00000000 01000001
boolean	1 byte	true/false	true	JVM内部实现，通常用 1 表示 true，0 表示 false
short	2byte			

3 变量存储

类型	存储位置	说明
局部变量	栈 Stack	每次调用方法就创建一个栈帧，局部变量存在其中
成员变量	堆 Heap	<code>new</code> 出来的对象 + 其字段
静态变量	方法区（元空间）	类加载时就分配，由 JVM 统一管理

4 引用类型是如何存储的？

例如：


```
Cat cat = new Cat();
```

- `cat` 是一个引用变量，**存在栈中**，保存的是一个地址
- 这个地址指向一个 `Cat` 对象，这个**对象存在堆中**
- 对象中的**成员变量也都存储在这个堆空间中**

引用变量存储的是地址，真实对象在堆中

5 所有数据最终都是“0/1”

无论是 int、float、char、对象、数组 —— 它们最终都会转换成二进制表示，再分配到内存的某个地址上

 **示例：int x = 100; 在内存中的表示（小端字节序）**

地址	内容（十六进制）
0x1000	64 (十六进制 0x64)
0x1001	00
0x1002	00
0x1003	00

100 的二进制是 00000000 00000000 00000000 01100100

你可以这样理解：

类型	存储在哪	存的是什么
基本类型 (int, float)	栈 / 堆	真实值的二进制
引用类型变量	栈	对象在堆中的地址
对象本身	堆	对象的字段值、类型信息等
数组	堆	数组长度 + 每个元素

小结：

1. 所有数据最终都是 0 和 1，在内存中以字节单位保存
2. 栈用于方法执行和局部变量；堆用于对象；方法区存静态类信息
3. 基本类型变量保存的是值本身，引用类型变量保存的是地址（引用）
4. JVM 管理内存的方式决定了变量的生命周期、访问速度和线程安全