

语言基础

变量

概念：计算机内的一块储存空间，是储存数据的基本单元。

```
数据类型 变量名 = 值;
```

变量的定义

- 声明

```
数据类型 变量名; int i;
```

- 赋值

```
变量名 = 值; i = 0;
```

- 注意：java是强类型语言，**变量的类型**必须与**数据的类型**一致。

变量定义的3种方式

1. 先定义再赋值

```
int i;  
i = 0;
```

2. 定义的同时并赋值

```
int i = 0;
```

3. 同类型、多变量同时定义、赋值。

```
int i,j,x,y= 1,2,3,4;
```

数据类型

- java的变量有严格的数据类型区分。
- 任何一个值都会有对应的数据类型。

8种基本数据类型

```
整数 byte short int long 浮点 float double 布尔 boolean 字符 char
```

引用数据类型

- 字符串 String
- 数组
- 对象

基本数据类型

1. 整数型

类型	字节	取值范围（二进制）	取值范围（十进制）
byte	1字节	$-2^7 \sim 2^7-1$	-128 ~ 127
short	2字节	$-2^{15} \sim 2^{15}-1$	-32768 ~ 32767
int	4字节	$-2^{31} \sim 2^{31}-1$	-2147483648 ~ 2147483647
long	8字节	$-2^{63} \sim 2^{63}-1$	-9223372036854775808 ~ 9223372036854775807

• 注意：int为整数的默认类型，如需为long类型赋值较大整数时，需在值的后面追加“L”

2. 浮点型

类型	字节	负数取值范围	正数取值范围
float	4字节	$-3.4\text{E}+38 \sim -1.4\text{E}-45$	$1.4\text{E}-45 \sim 3.4\text{E}+38$
double	8字节	$-1.7\text{E}+308 \sim -4.9\text{E}-324$	$4.9\text{E}-324 \sim 1.7\text{E}+308$

- 浮点型数值采用科学计数法表示：
 - 2E3 等价于 $2 * 10^3$ （结果：2000.0）
 - 3E5 等价于 $3 * 10^5$ （结果：300000.0）
- 注意：double为浮点数的默认类型，如需为float类型赋值时，需要在值的后面追加“F”

3. 布尔型

类型	字节	范围	描述
boolean	1字节	true/false	真/假

- boolean不参与算数运算。

4. 字符型

类型	字节	范围	描述
char	2字节	0~65535	unicode字符集

- unicode中每一个字符都对应着一二进制整数，可以使用多种方式赋值。

```
char c1 = 'A';
char c2 = 65 ;
char c3 = '\u0041';//16进制数对应的字符赋值
```

转义字符

转义字符	描述
\n	换行符
\t	制表符
\	反斜线
'	单引号
"	双引号

引用数据类型

```
String str = "字符串"
```

- 以下哪些赋值语句可以通过编译?
 - byte a = 128; × 超出取值范围
 - short b = 65; ✓
 - short c = -32000; ✓
 - float d = 12.34; × 浮点数默认为double类型，此时应加F或f
 - char e = '65'; × 字符只能有一个
 - char f = 65; ✓ 可直接赋值为整数
 - short g = 65; char h = g; × short g可能为负数，无法直接赋值给char h
 - boolean i = "true"; × true / false 为Java保留字，直接使用
 - String j = " 123"; ✓ 空格也是字符

类型转换

自动类型转换

- 两种类型相互兼容
- 目标类型大于源类型

```
short s = 10;  
int i = s;
```

强制类型转换

- 两种类型相互兼容
- 目标类型小于源类型

```
short s = 12;  
byte n = (byte) s;
```

强转规则

- 整数长度足够，数据完整。
例：int i = 100; byte b = (byte)i; //b = 100
- 整数长度不够，数据截断。
例：int i = 10000; byte b = (byte)i; //b = 16 (符号位变化，可能变为负数)
- 小数强转整数，数据截断。
例：double d = 2.5; int i = (int)d; //i = 2 (小数位舍掉)
- 字符整数互转，数据完整。
例：char c = 65; int i = c; //i = 65
- boolean的取值为true/false，不可与其他类型转换。

自动类型提升

- 进行算数运算时：
 - 两个操作数有一个为double，计算结果提升为double。
 - 如果操作数中没有double，有一个为float，计算结果提升为float。
 - 如果操作数中没有float，有一个为long，计算结果提升为long。
 - 如果操作数中没有long，有一个为int，计算结果提升为int。
 - 如果操作数中没有int，均为short或byte，计算结果仍旧提升为int。
- 特殊：任何类型与String相加 (+) 时，实为拼接，其结果自动提升为String。

运算符

- 算数运算符

• `+` `-` `*` `/` `%`

- 一元运算符

`++` `--`

- 赋值运算符

`=` `+=` `-=` `*=` `/=` `%=`

- 关系运算符

`=` `>` `<` `>=` `<=` `==` `!=`

- 逻辑运算符

`&&` `||` `!`

- 三元运算符

布尔表达式? 结果一:结果二

控制台输入

```
Scanner scanner = new Scanner(System.in);  
String str = scanner.next();
```