

# 04 Java 基础

- 1 注释：对代码的解释说明
  - 单行注释 // 注释信息
  - 多行注释 /\* 注释信息 \*/
  - 文档注释 /\*\* 注释信息 \*/
  - 不要嵌套注释
- 2 关键字：被Java赋予特定涵义的英文单词
  - 特点
    - 字母全是小写
    - 在代码编辑器中，关键字均是特殊颜色的标记。
  - 2.1 class: 用于创建/定义一个类，后面跟随类名。
    - (类是Java最基本的组成单元)
    - 注意：类名和文件名一定保持一致
- 3 字面量：数据在程序中的书写格式
  - 类型：整数、小数、字符串、字符、布尔、空类型

字面量类型	说明	举例
整数类型	不带小数点的数字	666, -88
小数类型	带小数点的数字	13.14, -5.21
字符串类型	用双引号括起来的内容	"HelloWorld", "黑马程序员"
字符类型	用单引号括起来的,内容只能有一个	'A', '0', '我'
布尔类型	布尔值, 表示真假	只有两个值: true, false
空类型	一个特殊的值, 空值	值是: null

- 注意1：凡是用双引号括起来的都是字符串类型。哪怕里面只有一个字符或者什么都不写
- 注意2：null不能直接打印。如果需要打印null，那么只能用字符串的形式进行打印。  
System.out.println("null");
- 注意3：定义字符串是：String 变量名 = "xxx";
- 4 特殊的字符类型的字面量
  - \t 制表符：
    - 在打印时，把前面字符串的长度补齐到8，或者8的整数倍。最少补一个空格，最多补8个空格。

```
public class HelloWorld {
    public static void main(String[] args) {
        System.out.println("abc" + '\t');
    }
}
```

*(Handwritten note: "abc" + '\t' is followed by a red "5" with a plus sign, indicating 5 spaces are added.)*

- 此处，abc 3个字符，所以 \t 占5个空格

```
public class HelloWorld {  
    public static void main(String[] args) {  
        System.out.println("name" + '\t' + "age");  
        System.out.println("tom" + '\t' + "23");  
    }  
}
```

```
name    age  
tom     23
```

- 制表符使得数据对齐

## • 5 变量

- 定义格式：数据类型 变量名 = 数据值；

数据类型	关键字
整数	int
浮点数	double

- 浮点数就是小数
- 注意事项：
  - 1.只能存一个值
  - 2.变量名不能重复定义
  - 3.一条语句可以定义多个变量    int a=100, b=200, c=300;
  - 4.变量在使用之前一定要进行赋值 int d; 还需要将数据赋给他
  - 5.变量的作用域范围

## • 6 数据类型

- 分为基本数据类型和引用数据类型

数据类型	关键字	取值范围	内存占用
整数	byte	-128~127	1
	short	-32768~32767	2
	int	-2147483648~2147483647 (10位数)	4
	long	-9223372036854775808 ~ 9223372036854775807 (19位数)	8
浮点数	float	-3.401298e-38 到 3.402823e+38	4
	double	-4.9000000e-324 到 1.797693e+308	8
字符	char	0-65535	2
布尔	boolean	true, false	1

- 如果要定义long类型的变量，需要在数据值后面加上一个L作为后缀。例：long n = 9999999L;
- 如果要定义float类型的变量，需要在数据值后面加上一个F作为后缀。例：float f = 10.1F;
- 取值范围:double>float>long>int>short>byte

## • 7 标识符

- 定义：给类、方法、变量等起的名字
- 标识符命名规则
  - 由数字、字母、下划线和美元符\$组成
  - 不能以数字开头
  - 不能是关键字
  - 区分大小写
- 建议：
  - 小驼峰命名：适用于方法、变量
    - 标识符是一个单词时，全部小写 **name**
    - 标识符由多个单词组成时，第一个单词首字母小写，其他单词首字母大写 **firstName**
  - 大驼峰命名：适用于类名
    - 标识符是一个单词时，首字母大写 **Student**
    - 标识符由多个单词组成的时候，每个单词的首字母大写 **GoodStudent**

## • 8 键盘录入

- Scanner这个类可以接收键盘输入 **整数**
- 步骤：
  - import java.util.Scanner; //导包
  - Scanner sc = new Scanner(System.in); //创建对象：sc为变量名，可以改变，其余不能变
  - int i = sc.nextInt(); //i为变量名，可以改变，其余不能变

## • 9 运算符

- 9.1 算术运算符

符号	作用
+	加
-	减
*	乘
/	除
%	取模、取余

- 整数参与计算，结果只能是整数
- 在代码中，有小数参与计算，结果有可能不精确
- 取余的应用场景
  - 1.判断A是否可以被B整除：A%B=0
  - 2.判断A是否为偶数：A%2=0
  - 3.斗地主发牌：三个玩家；把每一张牌都定义一个序号，拿序号%3，如果结果为1，发给第一个玩家；如果结果为2，发给第二个玩家；如果结果为0，发给第三个玩家
- 9.1.1 数值相加
  - 数字运算时，数据类型不一样不能运算，需要转成一样的，才能运算
    - 隐式转换：把取值范围小的数变成取值范围大的数
      - int a = 10; double b = a; 此时 b = 10.0
      - byte short char 三种类型在数据运算时，都会直接提升为 int，然后再进行计算
    - 强制转换：目标数据类型 变量名 = (目标数据类型) 被强转的数据；
      - double a = 12.3; int b =(int) a;
- 9.1.2 字符串相加
  - 当“+”操作中出现字符串时，这个“+”是字符串连接符，就不是算术运算符了。会将前后的数据进行拼接，并产生一个新的字符串。
    - “123”+123 输出“123123”
  - 连续进行“+”操作，从左到右逐个执行
    - 1+99+“年黑马” 输出“100年黑马”

```
System.out.println(3.7 + "abc");//"3.7abc"
System.out.println("abc" + true); //"abctrue"
System.out.println('中' + "abc" + true); //"中abctrue"

int age = 18;
System.out.println("我的年龄是" + age + "岁");//"我的年龄是18岁"
System.out.println("我的年龄是" + "age" + "岁");//"我的年龄是age岁"
```

### 9.1.3 字符相加

- 当字符+字符、字符+数字时，会把字符通过ASCII码表查询到对应的数字再进行计算。

高四位 低四位		ASCII非打印控制字符																ASCII 打印字符															
		0000								0001								0010	0011	0100	0101	0110	0111										
		0								1								2	3	4	5	6	7										
		十进制	字符	ctrl	代码	字符解释	十进制	字符	ctrl	代码	字符解释	十进制	字符	十进制	字符	十进制	字符	十进制	字符	十进制	字符	十进制	字符	十进制	字符	十进制	字符	十进制	字符	ctrl			
0000	0	0	BLANK 空白	^@	NUL	空	16	▶	^P	DLE	数据链路转意	32		48	0	64	@	80	P	96	`	112	p										
0001	1	1	☺	^A	SOH	标题开始	17	◀	^Q	DC1	设备控制 1	33	!	49	1	65	A	81	Q	97	a	113	q										
0010	2	2	☹	^B	STX	正文开始	18	↕	^R	DC2	设备控制 2	34	"	50	2	66	B	82	R	98	b	114	r										
0011	3	3	♥	^C	ETX	正文结束	19	!!	^S	DC3	设备控制 3	35	#	51	3	67	C	83	S	99	c	115	s										
0100	4	4	♦	^D	EOF	传输结束	20	↑	^T	DC4	设备控制 4	36	\$	52	4	68	D	84	T	100	d	116	t										
0101	5	5	♣	^E	ENQ	查询	21	⌂	^U	NAK	反确认	37	%	53	5	69	E	85	U	101	e	117	u										
0110	6	6	♠	^F	ACK	确认	22	■	^V	SYN	同步空闲	38	&	54	6	70	F	86	V	102	f	118	v										
0111	7	7	●	^G	BEL	震铃	23	↓	^W	ETB	传输块结束	39	'	55	7	71	G	87	w	103	g	119	w										
1000	8	8	□	^H	BS	退格	24	↑	^X	CAN	取消	40	(	56	8	72	H	88	X	104	h	120	x										
1001	9	9	○	^I	TAB	水平制表符	25	↓	^Y	EM	媒体结束	41	)	57	9	73	I	89	Y	105	i	121	y										
1010	A	10	◻	^J	LF	换行/换行	26	→	^Z	SUB	替换	42	*	58	:	74	J	90	Z	106	j	122	z										
1011	B	11	♂	^K	VT	垂直制表符	27	←	^[	ESC	转意	43	+	59	;	75	K	91	[	107	k	123	{										
1100	C	12	♀	^L	FF	换页/翻页	28	↵	^[	FS	文件分隔符	44	,	60	<	76	L	92	\	108	l	124											
1101	D	13	♂	^M	CR	回车	29	↔	^]	GS	组分隔符	45	-	61	=	77	M	93	]	109	m	125	}										
1110	E	14	♂	^N	SO	移出	30	▲	^_	RS	记录分隔符	46	.	62	>	78	N	94	^	110	n	126	~										
1111	F	15	☼	^O	SI	移入	31	▼	^-	US	单元分隔符	47	/	63	?	79	O	95	_	111	o	127	Δ										

## 9.2 自增自减运算符

符号	作用	说明
++	加	变量的值加1
--	减	变量的值减1

- ++和--可以放在变量前面或者后面
- int a = 10; int b = a ++; 此时a=11, b=10 先用后加
- int a = 10; int b = ++a; 此时a=11, b=11 先加后用

## 9.3 赋值运算符

符号	作用	说明
=	赋值	int a=10, 将10赋值给变量a
+=	加后赋值	a+=b, 将a+b的值给a
-=	减后赋值	a-=b, 将a-b的值给a
*=	乘后赋值	a*=b, 将a*b的值给a
/=	除后赋值	a/=b, 将a÷b的商给a
%=	取余后赋值	a%=b, 将a÷b的余数给a

- +=、-=、\*=、/=、%=都隐藏了一个强制类型转换

## 9.4 关系运算符

符号	说明
==	a==b, 判断a和b的值是否相等, 成立为true, 不成立为false
!=	a!=b, 判断a和b的值是否不相等, 成立为true, 不成立为false
>	a>b, 判断a是否大于b, 成立为true, 不成立为false
>=	a>=b, 判断a是否大于等于b, 成立为true, 不成立为false
<	a<b, 判断a是否小于b, 成立为true, 不成立为false
<=	a<=b, 判断a是否小于等于b, 成立为true, 不成立为false

- 注意：关系运算符的结果都是boolean类型，要不是true，要不是false。不要把“==”写成“=”

## 9.5 逻辑运算符

符号	作用	说明
&	逻辑与（且）	并且，两边都为真，结果才是真
	逻辑或	或者，两边都为假，结果才是假
^	逻辑异或	相同为 false，不同为 true
!	逻辑非	取反

- java中表示 $5 < x < 15$ 不能连着，应该 $5 < x \& x < 15$
- 短路运算符：效率高（当左边的表达式能确实最终的结果，右边的表达式就不会参与运行了）

符号	作用	说明
&&	短路与	结果和&相同，但是有短路效果
	短路或	结果和 相同，但是有短路效果

```
int a = 10;
int b = 10;
boolean result = ++a < 5 && ++b < 5;
System.out.println(result); // false
System.out.println(a); // 11
System.out.println(b); // 10
```

- 此处，先判断++a为11，&&左边为假，无需判断右边，所以不执行++b

## 9.6 三元运算符

- 格式： 关系表达式？ 表达式1： 表达式2；

```
int max = a > b ? a : b; 把三元运算符的结果赋值给一个变量
System.out.println(a > b ? a : b); 把三元运算符的结果直接打印
```

- 计算规则：
  - 先计算关系表达式的值。为true，表达式1的值为运算结果；否则表达式2的值为运算结果
  - 三元运算符的结果必须要被使用

## 9.7 运算符的优先级

优先级	运算符
1	. () {}
2	!, ~, ++, --
3	*, /, %
4	+, -
5	<<, >>, >>>
6	<, <=, >, >=, instanceof
7	==, !=
8	&
9	^
10	
11	&&
12	
13	?:
14	=, +=, -=, *=, /=, %=, &=,