Java基础语法





关于Java语言基础知识同学们需要学会什么

类型转换知识

运算符的使用

怎么调用Java提供的程序

类型的变量或者数据 互相传输,不同类型 及一些逻辑运算、 据的类型如何确定?

如何实现程序中不同 程序中怎么进行数据 的基本算术运算、以 的数据运算的结果数 Java提供了具体哪些 运算方式

程序员需要调用Java 写好的一些程序,这 些程序如何去调用

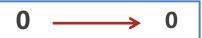


> 变量详解

- ◆ 变量里的数据在计算机中的底层原理
- ◆ 数据类型
- > 类型转换
- > 运算符

二进制

● 只有0、1,按照逢2进1的方式表示数据:



$$\begin{array}{c} 1 \\ +1 \\ \hline 10 \end{array}$$

$$\begin{array}{c}
10 \\
+1 \\
\hline
11
\end{array}$$

$$4 \longrightarrow \frac{11}{100}$$

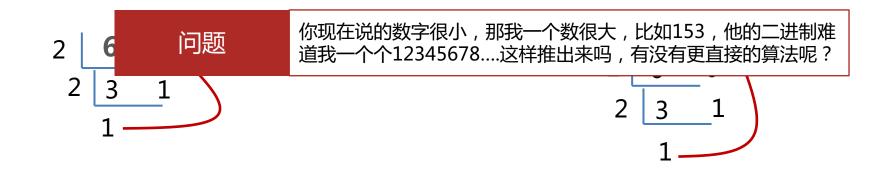
想一想

请问:6是的二进制是多少?



十进制转二进制的算法

• 除二取余法。





计算机中的数据的最小单位



- 计算机中数据最小的组成单元:使用8个二进制位为一组,称之为一个字节(byte,简称B)
- 字节中的每个二进制位就称为 位 (bit, 简称b), 1B = 8b





问题1

你跟我讲这么多,我听是听懂了,有什么用啊?

问题2

平时我们接触到的数据不只是数字啊,我们接触到的都是电影,图片、声音、文字啊,这些在计算机里是怎么存的啊?





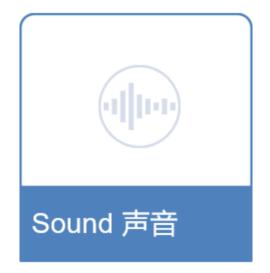
- 1、数据在计算机底层都是怎么存储的?
 - 都是采用二进制:使用0、1,按照逢2进1的规则表示数据来存储。
- 2、如何快速的算出一个数据的二进制形式?
- 3、数据在计算机底层存储的最小单位是什么?
 - 字节,一个字节等于8个二进制位:1B=8b



其他数据形式是如何存储的









字符在计算机中是如何存储的呢?

● ASCII编码表:即美国信息交换标准编码,规定了现代英语、数字字符、和其他西欧字符对应的数字编号。



char ch = 'a';

	We of 1 1 11 14 2 3 5 2																							
一層	四位	ASCII非打印控制字符						ASCII 打印字符																
/ -		0000			0001			0010 0011		0100 0101		0110		0111										
W. m. 42		0			1			2 3			4		5		6		7							
低四	1立 /	十進制	字符	ctrl	代码	字符解释	十進制	字符	ctrl	代码	字符解释	十進制	字符	十進制	字符	十進制	字符	十進制	字符	十進制	字符	十進制	字符	ctrl
0000	0	0	BLANK NULL	^@	NUL	空	16	•	^ P	DLE	数据链路转意	32		48	0	64	@	80	Р	96	`	112	р	
0001	1	1	0	^ A	SOH	头标开始	17	•	^Q	DC1	设备控制 1	33	ļ.	49	1	65	Α	81	Ø	97	а	113	q	
0010	2	2	•	^ B	STX	正文开始	18	1	^R	DC2	设备控制 2	34	"	50	2	66	В	82	R	98	b	114	r	
0011	3	3	٧	^c	ETX	正文结束	19	!!	^ S	DC3	设备控制 3	35	#	51	3	67	С	83	S	99	С	115	s	
0100	4	4	٠	^D	EOT	传输结束	20	¶	^ T	DC4	设备控制 4	36	\$	52	4	68	D	84	Т	100	d	116	t	
0101	5	5	*	^ E	ENQ	查询	21	∮	^ U	NAK	反确认	37	%	53	5	69	Е	85	U	101	е	117	u	
0110	6	6	٨	^F	ACK	确认	22		^ V	SYN	同步空闲	38	&	54	6	70	F	86	٧	102	f	118	V	
0111	7	7	•	^ G	BEL	震铃	23	1	^ W	ЕТВ	传输块结束	39	•	55	7	71	G	87	w	103	g	119	W	
1000	8	8		^н	BS	退格	24	1	^X	CAN	取消	40	(56	8	72	Н	88	Х	104	h	120	Х	
1001	9	9	0	^Ι	TAB	水平制表符	25	↓	^ Y	EM	媒体结束	41)	57	9	73	I	89	Υ	105	i	121	У	
1010	A	10	0	^ј	LF	换行/新行	26	\rightarrow	^ Z	SUB	替换	42	*	58	:	74	J	90	Z	106	j	122	z	
1011	В	11	ď	^ K	VT	竖直制表符	27	←	1 ^	ESC	转意	43	+	59	;	75	Κ	91	[107	k	123	{	
1100	С	12	Q	^L	FF	换页/新页	28	_	^\	FS	文件分隔符	44	,	60	<	76	L	92	1	108	T	124	Ī	
1101	D	13	₽	^ M	CR	回车	29	\leftrightarrow	^]	GS	组分隔符	45	-	61	=	77	М	93]	109	m	125	}	
1110	Е	14	A	^ N	SO	移出	30	A	^6	RS	记录分隔符	46		62	>	78	Ν	94	^	110	n	126	٧	
1111	F	15	¤	^0	SI	移入	31	▼	^-	US	单元分隔符	47	1	63	?	79	0	95		111	0	127	Δ	^Back space

ASCII 字符代码表 一



图片数据-彩色图

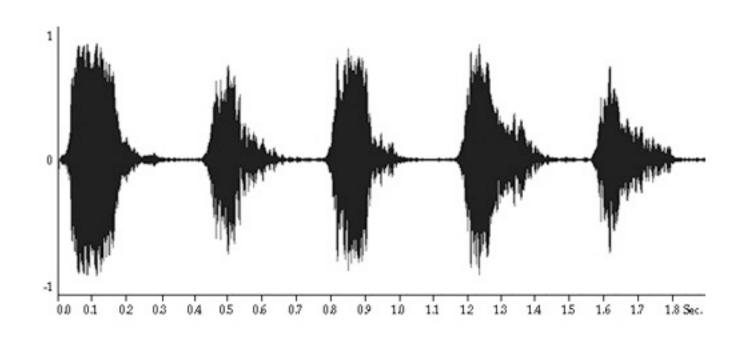
- 图片就是无数个像素点组成的
- 每个像素点的数据:用0~255*255*255表示其颜色

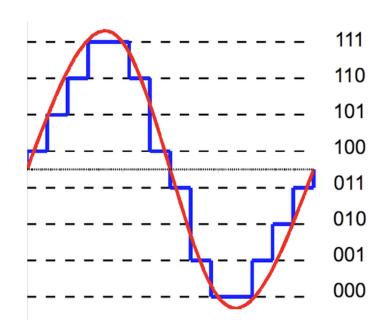




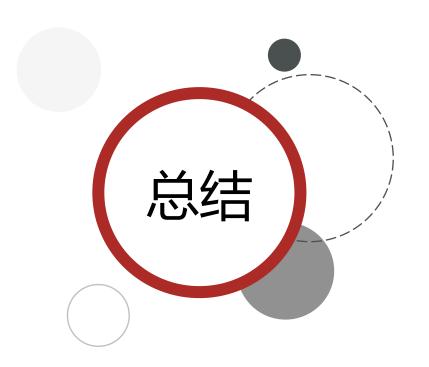


声音数据



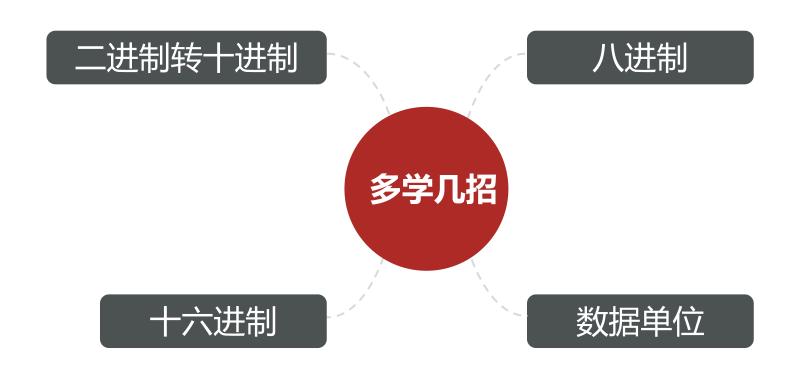






- 1、字符数据在计算机中是怎么存的?
 - 字符存的是ascii码表中对应的数字的二进制形式。
 - 字符'A'对应的数字是 65
 - 字符'a'对应的数字是 97
 - 字符'0'对应的数字是 48
- 2、图片和音视频等文件的数据是怎么存储的啊?
 - 也是采用二进制进行存储的。





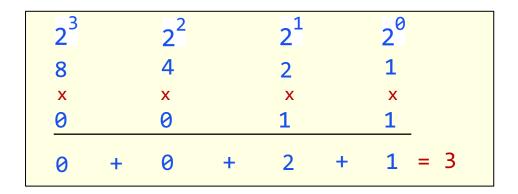


十进制转二进制的算法

• 十进制数转二进制数:除二取余法

结果:13的二进制是1101

● 二进制数转十进制数



103	10 ²		10 ¹		100	
1000	100		10		1	
X	X		X		X	
1	9		9		8	
1000 +	900	+	90	+	8 =	1998

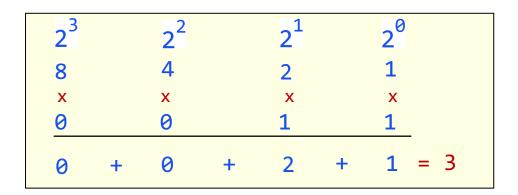


十进制转二进制的算法

• 十进制数转二进制数:除二取余法

结果:13的二进制是1101

● 二进制数转十进制数





八进制、十六进制介绍

- 为了便于观察和表示二进制,推出了八进制和十六进制。
- 每3位二进制作为一个单元,最小数是0,最大数是7,共8个数字,这就是八进制。

97: 01100001

01, 100, 001 --> 141

● 每4位二进制作为一个单元,最小数是0,最大数是15,共16个数字,依次用: 0~9ABCDEF代表就是十六进

97: 01100001

0110, 0001 --> 61

250: 11111010

1111、1010 --> FA

注意

Java程序中支持书写二进制、八进制、十六进制的数据,分别需要以OB或者Ob、O、OX或者Ox开头。



计算机的数据单位

- 计算机最小的组成单元是:字节,1B=8b。
- 在B的基础上,计算机发展出了KB、MB、GB、TB、.... 这些数据单位。

$$1B = 8b$$

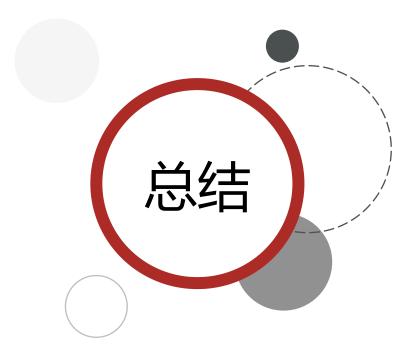
1KB = 1024B

1MB = 1024KB

1GB = 1024MB

1TB = 1024GB





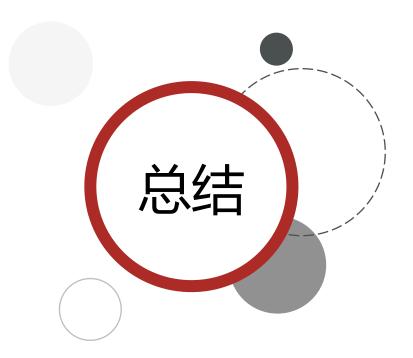
1. 二进制如何计算成十进制?

2	3	2 ²		20	
8		4		1	
x		X		x	
1		1	0	1	
8	+	4		+ 1	= 13

- 2. 二进制如何计算成八进制?
 - 每3位二进制作为一个单元,最小数是0,最大数是7,0-7有8个数字。

二进制转八进制: 01100001 ---> 001 100 001 -> 141





3. 二进制如何计算成十六进制?

- 每4位二进制作为一个单元,最小数是0,最大数是15
- 0-15有16个数字, 依次用: 0~9 A B C D E F 代表

二进制转十六进制:

01100001 ---> 0110 0001 -> 61

11111010 ---> 1111 1010 -> FA

4. 数据大小的单位换算是怎么样的?

$$1B = 8b$$

1KB = 1024B

1MB = 1024KB

1GB = 1024MB

1TB = 1024GB



> 变量详解

- ◆ 变量里的数据在计算机中的底层原理
- ◆ 数据类型
- > 类型转换
- > 运算符



数据类型的作用

● 数据类型就是约束变量存储数据的形式。

数据类型 变量名称 = 初始值;

```
public class HelloWorld {
    public static void main(String[] args) {
        double money = 9.9;

    int age = 23;
}
}
```







数据类型的分类

- 引用数据类型(除基本数据类型之外的,如String,其他的后面学习)
- 基本数据类型:4大类8种。

数据类型	关键字	取值范围	内存占用(字节数)
	byte	-128~127	1
	short	-32768~32767	2
整数	int(默认)	-2147483648~2147483647 (10位数)	4
	long	-9223372036854775808 ~ 9223372036854775807 (19位数)	8
~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~~	float	1.401298e-45到3.402823e+38	4
浮点数	double (默认)	4.900000e-324 到1.797693e+308	8
字符	char	0-65535	2
布尔	boolean	true, false	1





#### 1. 数据类型分为几种?

- 引用数据类型:String
- 基本数据类型:4大类8种。
  - ✓ byte short int (默认) long 整型
  - ✓ float double (默认) 浮点型
  - ✓ char 字符型
  - ✓ boolean 布尔型
- 2. 随便写的整数、小数字面值默认什么类型?
  - 23 , 默认是int类型 , 加上L/l就是long类型的数据了。
  - 23.8,默认是double类型,加上F/f就是float类型了。



- > 变量详解
- > 类型转换
  - ◆ 自动类型转换
  - ◆ 表达式的自动类型转换
  - ◆ 强制类型转换
- > 运算符



# 什么是自动类型转换,为什么要学习自动类型转换?

◆ 类型范围小的变量,可以直接赋值给类型范围大的变量。

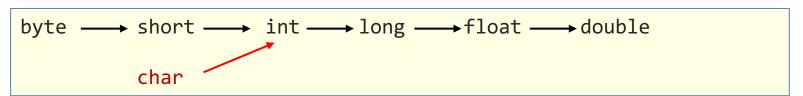
# 自动类型转换的底层原理

```
byte a = 12 ;
int b = a;
System.out.println(b); // 12
```

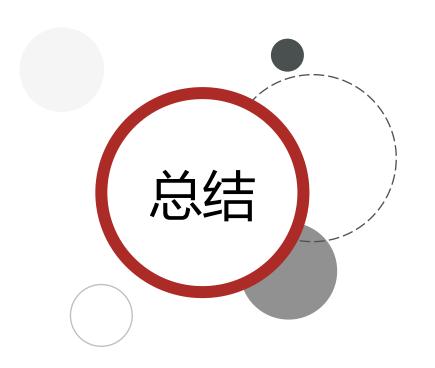
a | 00001100 | _(8位)

b 00000000 00000000 0000000 00001100 (32位

# 自动类型转换的其他形式







- 1. 为什么要进行类型转换?
  - 存在不同类型的变量赋值给其他类型的变量
- 2. 自动类型转换是什么样的?
  - 类型范围小的变量,可以直接赋值给类型范围大的变量。



- > 变量详解
- > 类型转换
  - ◆ 自动类型转换
  - ◆ 表达式的自动类型转换
  - ◆ 强制类型转换
- > 运算符
- > 案例知识:键盘录入技术



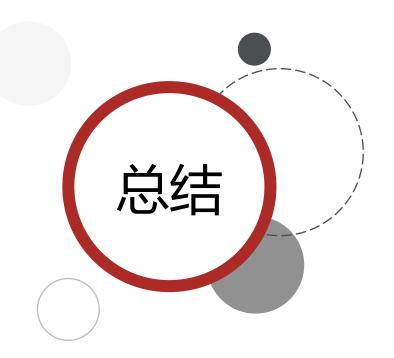
# 表达式的自动类型转换

● 在表达式中,小范围类型的变量会自动转换成当前较大范围的类型再运算。

# 注意事项:

- 表达式的最终结果类型由表达式中的最高类型决定。
- 在表达式中, byte、short、char 是直接转换成int类型参与运算的。





- 1. 表达式的自动类型转换是什么样的?
  - 小范围的类型会自动转换成大范围的类型运算。
- 2. 表达式的最终结果类型是由谁决定的?
  - 最终类型由表达式中的最高类型决定。
- 3. 表达式的有哪些类型转换是需要注意的?
  - byte short char是直接转换成int类型参与运算的。



- > 变量详解
- > 类型转换
  - ◆ 自动类型转换
  - ◆ 表达式的自动类型转换
  - ◆ 强制类型转换
- > 运算符
- > 案例知识:键盘录入技术



# 场景

```
int a = 功能1();
功能2(byte b);
```

```
int a = 20;
byte b = a; //报错
```

# 问题

● 类型范围大的数据或者变量,不能直接**赋值**给类型范围小的变量,会报错。

# 强制类型转换

● 可以强行将类型范围大的变量、数据赋值给类型范围小的变量。

数据类型 变量2 = (数据类型)变量1、数据

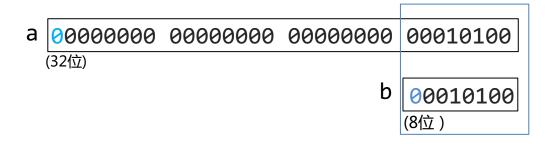
```
int a = 20;
byte b = (byte)a;
```



# 强制类型转换底层原理

```
int a = 20;
byte b = (byte)a;
System.out.println(b); // 20
```

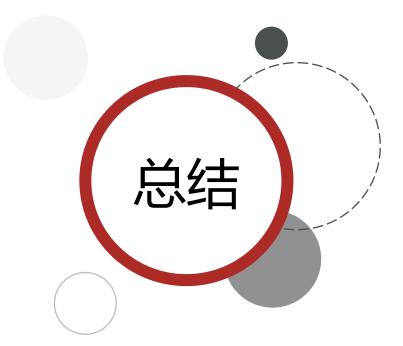
```
int i = 1500;
byte j = (byte)i;
System.out.println(j); // -36
```



# 注意事项

- 强制类型转换**可能**造成数据(丢失)溢出;
- 浮点型强转成整型,直接丢掉小数部分,保留整数部分返回。





- 1. 什么是强制类型转换?
  - 可以强行将类型范围大的变量、数据赋值给类型范围小的变量。
  - 数据类型 变量 = (数据类型)变量、数据

- 2. 强制类型转换有哪些需要注意的?
  - 可能出现数据丢失。
  - 小数强制转换成整数是直接截断小数保留整数。



- > 变量详解
- > 类型转换
- > 运算符
  - ◆ 基本算数运算符
  - ◆ +符号做连接符
  - ◆ 自增自减运算符
  - ◆ 赋值运算符
  - ◆ 关系运算符
  - ◆ 逻辑运算符
  - ◆ 三元运算符
  - ◆ 运算符优先级
- > 案例知识:键盘录入技术



# 运算符

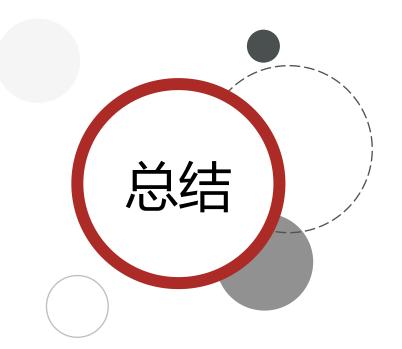
● 运算符:对字面量或者变量进行操作的**符号。** 

# 算数运算

符

付								
符号	作用	。 · · · · · · · · · · · · · · · · · · ·						
+	巾	参考小学一年级						
-	减	参考小学一年级						
*	乘	参考小学二年级,与 "×"相同						
/	除	与"÷"相同,注意:在Java中两个整数相除结果还是整数。						
%	取余	获取的是两个数据做除法的 <mark>余数</mark>						





1. 算数运算符有哪些?

• + · - · * / · %

- 2. / 需要注意什么, 为什么?
  - 如果两个整数做除法,其结果一定是整数,因为最高类型是整数。



# **国** 案例

## 数值拆分

需求:一个三位数,将其拆分为个位、十位、百位后,打印在控制台

3 2 1

②:个位的计算:数值%10

公式总结:

个位 : 数值 % 10

十位 : 数值 / 10 % 10

百位 : 数值 / 10 / 10 % 10

千位 : 数值 / 10 / 10 / 10 % 10;

• • •



- > 类型转换
- > 运算符
  - ◆ 基本算数运算符
  - ◆ +符号做连接符
  - ◆ 自增自减运算符
  - ◆ 赋值运算符
  - ◆ 关系运算符
  - ◆ 逻辑运算符
  - ◆ 三元运算符
  - ◆ 运算符优先级
- > 案例知识:键盘录入技术



#### 做连接符

"+"符号与字符串运算的时候是用作连接符的,其结果依然是一个字符。

#### 盅

```
int a = 5:
System. out. println("abc" + 'a');
System.out.println("abc" + a);
System.out.println(5 + a);
System. out.println("abc" + 5 + 'a');
System. out. println(15 + "abc" + 15);
System. out. println(a + 'a');
System. out. println(a + "" +'a');
System. out.println(a + 'a'+" itheima ");
System. out.println("itheima"+ a + 'a');
System. out. println("itheima"+ (a + 'a'));
```

【支付宝】支付宝 251*@qq.* 花 呗 05月 月账单 13839.17元,还 款日 05月20日, 你还有11个花呗 金可用于抵扣还款

#### 独门秘籍:

能算则算,不能算就在一起。(计算机很聪明)





- 1. + 除了做基本数学运算,还有哪些功能?
  - 与字符串做+运算时会被当成连接符,其结果还是字符串。
  - 能算则算,不能算就在一起。



- > 类型转换
- > 运算符
  - ◆ 算数运算符
  - ◆ +符号做连接符
  - ◆ 自增自减运算符
  - ◆ 赋值运算符
  - ◆ 关系运算符
  - ◆ 逻辑运算符
  - ◆ 三元运算符
  - ◆ 运算符优先级
- > 案例知识:键盘录入技术

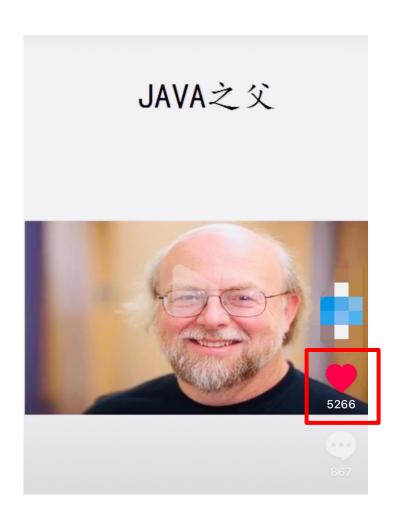


#### 自增自减运算符

符号	作用	说明
++	自增	变量自身的值加1
	自减	变量自身的值减1

#### 注意:

- ++ 和 -- 既可以放在变量的后边,也可以放在变量的前边。
- ++、--只能操作变量,不能操作字面量的。





#### 自增自减的使用注意事项

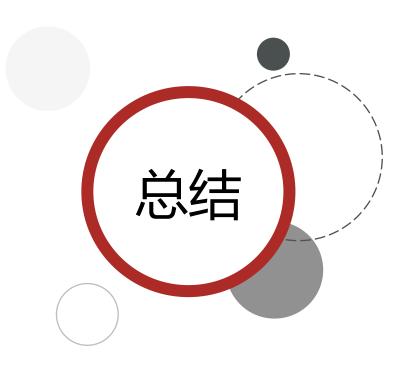
- ++、--如果不是单独使用(如在表达式中、或者同时有其它操作),放在变量前后会存在明显区别
  - ▶ 放在变量的前面, 先对变量进行+1、-1, 再拿变量的值进行运算。

```
int a = 10;
int rs = ++a;
```

▶ 放在变量的后面, 先拿变量的值进行运算, 再对变量的值进行+1、-1

```
int b = 10;
int rs = b++;
```





- 1. 自增、自减运算符是什么,有什么作用,需要注意什么?
  - ++、--。对当前变量值+1、-1
  - 只能操作变量,不能操作字面量
- 2. 自增、自减运算符在变量前后有区别吗?
  - 如果单独使用放前后是没有区别的。
  - 非单独使用:在变量前,先进行变量自增/自减,再使用变量

● 非单独使用:在变量后,先使用变量,再进行变量自增/自减

•





## 自增、自减拓展案例(有些面试题的笔试题可能会出现)

```
int c = 10;
int d = 5;
int rs3 = c++ + ++c - --d - ++d + 1 + c--;
System.out.println(rs3);
System.out.println(c);
System.out.println(d);
```

### > 类型转换

### > 运算符

- ◆ 算数运算符
- ◆ +符号做连接符
- ◆ 自增自减运算符
- ◆ 赋值运算符
- ◆ 关系运算符
- ◆ 逻辑运算符
- ◆ 三元运算符





#### 基本赋值运算符

● 就是 "="。

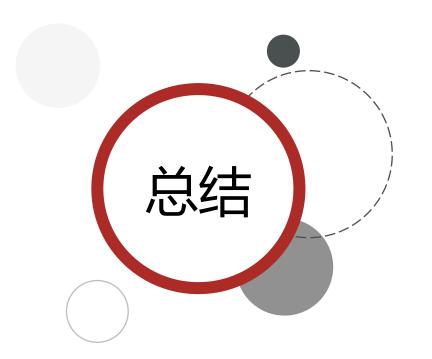
int a = 10; // 先看"="右边,把数据10赋值给左边的变量a存储。

#### 扩展赋值运算符

符号	作用	说明
+=	加后赋值	a+=b 等价于 a = (a的数据类型)(a+b); 将a + b的值给a
-=	减后赋值	a-=b 等价于 a = (a的数据类型)(a-b); 将a - b的值给a
*=	乘后赋值	a*=b 等价于 a = (a的数据类型)(a*b); 将a * b的值给a
/=	除后赋值	a/=b 等价于 a = (a的数据类型)(a/b); 将a / b的商给a
%=	取余后赋值	a%=b 等价于 a = (a的数据类型)(a%b); 将a % b的商给a

注意:扩展的赋值运算符隐含了强制类型转换。





1. 赋值运算符有哪些?

• =, +=, -=, *=, /=, %=

- 2. 作用是什么?
  - 基本作用是赋值,扩展运算符还自带了强制类型转换。
  - +=还可以实现数据的累加,把别人的数据加给自己。



- > 类型转换
- > 运算符
  - ◆ 算数运算符
  - ◆ +符号做连接符
  - ◆ 自增自减运算符
  - ◆ 赋值运算符
  - ◆ 关系运算符
  - ◆ 逻辑运算符
  - ◆ 三元运算符
  - ◆ 运算符优先级
- > 案例知识:键盘录入技术



#### 关系运算符

● 是对数据进行条件判断的符号,最终会返回一个比较的布尔结果(false,true)。

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 <mark>不相等</mark> ,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 <mark>大于等于</mark> 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< th=""></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:在做相等判断是,干万不要把 == 误写成 =



- > 类型转换
- > 运算符
  - ◆ 算数运算符
  - ◆ +符号做连接符
  - ◆ 自增自减运算符
  - ◆ 赋值运算符
  - ◆ 关系运算符
  - ◆ 逻辑运算符
  - ◆ 三元运算符
  - ◆ 运算符优先级
- > 案例知识:键盘录入技术



#### 逻辑运算符概述

● 可以把多个条件的布尔结果放在一起运算,最终返回一个布尔结果。

全部结果 > 屏幕尺寸: 6.95英寸及以上 × 运行内存: 8GB × "手机"

符号	介绍	说明
&	逻辑与	必须都是true , 结果才是true; <mark>只要有一个是false , 结果一定是false。</mark>
I	逻辑或	只要有一个为true、结果就是true
!	逻辑非	你真我假、你假我真。 !true=false 、 !false= true
^	逻辑异或	如果两个条件都是false或者都是true则结果是false。两个条件不同结果是true。



#### 短路逻辑运算符

符号	介绍	说明
&&	短路与	判断结果与"&"一样。过程是 <b>左边为</b> false , 右边则不执行。
П	短路或	判断结果与" "一样。过程是 <b>左边为 true, 右边则不执行</b> 。

注意:逻辑与 "&" 、逻辑或 "|": 无论左边是 false还是 true , 右边都要执行。



#### 1. 逻辑运算符有哪些,有什么特点?



● 逻辑与 "&" :有一个为false、结果是false

● 短路与&&: 一个为false、结果是false。前一个为false,后一个条件不执行

了

● 逻辑或 "<mark>"</mark>:有一个为true、结果是true

● 短路或||:一个为true、结果是true。前一个为true,后一个条件不执行了

● 逻辑非 "!" :!false=true、!true=false

● 逻辑异或 "^" :相同是false、不同是true。

● 注意:实际开发中、常用的逻辑运算符还是:&&、||、!



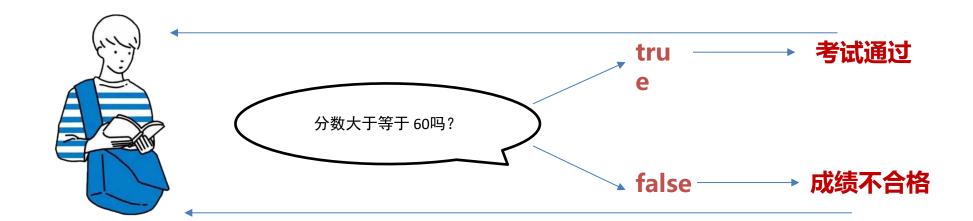
- > 类型转换
- > 运算符
  - ◆ 算数运算符
  - ◆ +符号做连接符
  - ◆ 自增自减运算符
  - ◆ 赋值运算符
  - ◆ 关系运算符
  - ◆ 逻辑运算符
  - ◆ 三元运算符
  - ◆ 运算符优先级
- > 案例知识:键盘录入技术



### 三元运算符介绍

● 格式: 条件表达式? 值1:值2;

● 执行流程:首先计算关系表达式的值,如果值为true,返回值1,如果为false,返回值2。





# 1 案例

## 求三个整数的最大值

需求:定义三个整数,找出最大值并打印在控制台。

#### 分析:

- ① 用三元运算符获取前两个整数的最大值,并用临时变量保存起来。
  - num1 > num2 ? num1 : num2;
- ② 用三元运算符,让临时最大值,和第三个整数,进行比较,并记录结果。
  - temp > num3 ? temp : num3;
- ③ 输出结果

- > 变量详解
- > 类型转换
- > 运算符
  - ◆ 算数运算符
  - ◆ +符号做连接符
  - ◆ 自增自减运算符
  - ◆ 赋值运算符
  - ◆ 关系运算符
  - ◆ 逻辑运算符
  - ◆ 三元运算符
  - ◆ 运算符优先级





#### 运算符优先级

● 在表达式中,哪个运算符先执行后执行是要看优先级的,例如"*、/"的优先级高于"+、-"。

优先级	运算符
1	0
2	!, ~, ++,
3	* . / . %
4	+、-
5	<<, >>, >>>
6	< 、 <= 、 >、 >=、 instanceof
7	== 、!=
8	&
9	^
10	I
11	8.8.
12	II
13	?:
14	=、 += 、 -= 、 *= 、 /=、 %=、 &=、





#### 运算符优先级

//目标:看看Java是否存在优先级,大家以后需要注意优先级问题

System. out. println(10 > 3 || 10 > 3 && 10 < 3); // true

System. *out*.println( (10 > 3 || 10 > 3 ) && 10 < 3); // false



传智教育旗下高端IT教育品牌