

## JAVA基础语法

回顾：

### 今天任务

1. 标识符
2. 标识符的命名规范
3. 关键字和保留字
4. 变量
5. 基本数据类型
6. 运算符

### 教学目标

1. 掌握什么是标识符
2. 标识符的命名规范
3. 掌握关键字和保留字
4. 掌握变量
5. 掌握八种基本数据类型
6. 掌握运算符

## 第一节：标识符、标识符的命名规范、关键字、保留字

### 1.1 什么是标识符

Java对包、类、方法、参数和变量等要素命名时使用的字符序列称为标识符。

### 1.2 标识符的命名规则

1. 由字母（含英、中、日、俄等）、数字、下划线（`_`）和美元符号（`$` 音：dollar）组成。
2. 不能以数字开头。
3. 区分大小写。
4. 长度无限制。（一般不要超过15个字符）
5. 不能是Java中的保留字和关键字。

### 1.3 标识符命名习惯

1. 标识符命名习惯：见名知意。
2. 使用驼峰式命名：  
类名：单词首字母大写，多个单词组成每个单词首字母大写  
变量名，方法名，参数名：首字母小写，如果多个单词组成，第一个单词首字母小写，剩下的单词首字母大写

### 1.4 关键字、关键字

关键字：Java中有一些赋予特定的含义，有专门用途的字符串称为关键字（keyword）。全部为小写  
保留字：没有定义用途，但保留备用。goto、const

关键字					
abstract	boolean	break	byte	case	catch
char	class	const	continue	default	do
double	else	extends	final	finally	float
for	goto	if	implements	import	instanceof
int	interface	long	native	new	package
private	protected	public	return	short	static
strictfp	super	switch	synchronized	this	throw
throws	transient	try	void	volatile	while
assert	enum				

## 第二节：八种基本数据类型

### 2.1 基本数据类型：

布尔型---- boolean

字符型---- char

整数型---- byte, short, int, long

浮点型---- float, double

### 2.2 基本数据类型的存储

Java中所有的基本数据类型都有固定的存储范围和所占内存空间的大小，而不受具体操作系统的影响，以保证Java程序的可移植性

### 2.3 整数型

Java语言的整型数值默认为int型，如：int i = 3;

要声明long型可以加'l'或'L'，如：long l = 3L;

注意：在数值超过int的表述范围时(2147483647)，需要加上l/L对long赋值

### 2.4 浮点型

Java浮点类型常量有两种表示形式

十进制数形式，必须含有小数点，例如：

3.14      314.0      0.314

科学记数法形式，如

3.14e2      3.14E2      314E2

Java浮点型常量默认为double型，如要声明一个常量为float型，则需在数字后面加f或F，如：

double d = 3.14; float f = 3.14f;

### 2.5 字符型

char型数据用来表示通常意义上“字符”

```
char c = 'A'; char c2 = '好';
```

Java字符采用Unicode编码，每个字符占两个字节，因而可用十六进制编码形式表示(Unicode是全球语言统一编码)

```
char c1 = '\u0061';//相当于'a'
```

```
char c2=97;//ASCII美国标准信息交换码，相当于'a'
```

Java语言中还允许使用转义字符'\''来将其后的字符转变为其它的含义：

```
char c2 = '\n';
```

## 转义字符

	名称	描述
\n	换行	将光标移到下一行的第一格。
\r	回车	将光标移到当前行的第一格。在命令行下会覆盖写。
\t	水平制表	将光标移到下一个水平制表位置。
\'	单引号	产生一个单引号。
\"	双引号	产生一个双引号。
\\	斜杠	产生一个斜杠。

// \ 转义字符 可以将后面的n 转义成为 换行的操作

```
System.out.print("i的值= "+i+"\n");
```

```
System.out.print("y的值 = "+y);
```

```
System.out.print("f的值 = "+f);
```

```
System.out.println("c的值 = "+c);
```

// \t 水平制表符

```
System.out.print("99\t98\t97\t96");
```

```
System.out.print("你好\你好");
```

```
System.out.print("你好\\你好");
```

## 2.6 布尔类型

boolean类型适于逻辑运算，一般用于程序流程控制

boolean类型数据只允许取值true或false，不可以用0或非0的整数替代true和false。

示例：

```
boolean b = false;
```

```
boolean c = true;
```

## 2.7 JAVA数据类型总结

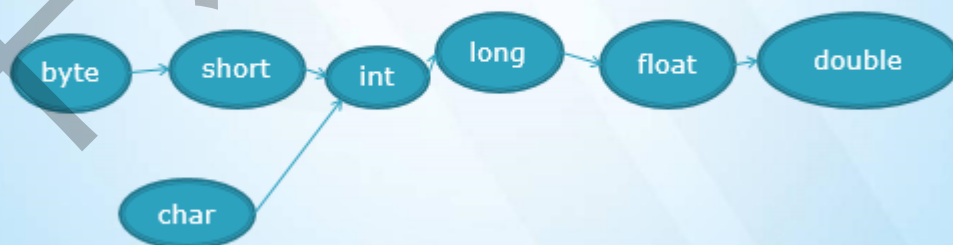
# JAVA基本数据类型总结

NO.	数据类型	大小/位	默认值 (全局变量才有)	可表示的数据范围
1	long(长整数)	64	0l	-9223372036854775808 ~9223372036854775807
2	int(整数)	32	0	-2147483648~2147483647
3	short(短整数)	16	(short) 0	-32768~32767
4	byte(字节型)	8	(byte) 0	-128~127
5	char(字符)	16	\u0000 0(空)	0-65535
6	float(单精度)	32	0.0f	-3.4E38~3.4E38
7	double(双精度)	64	0.0d	-1.7E308~1.7E308
8	boolean		false	true/false

## 2.8 数据类型的转换

1. `boolean` 类型不能转换成任何其它基本数据类型。
2. 自动类型转换：数据类型取值范围小的转为取值范围大的  
`byte, short, int, long, float, double`  
`char -> int, long, float, double`  
`short, char` 不会互相转换  
`byte, short, char` 三者 在计算时会转换成 `int` 类型（练习）
3. 强制类型转换：数据类型取值范围大的 转为 取值范围小的，需要加强制转换符，也就是（要转换的数据类型）。  
`long l = 100L; int i = (int)l;`  
但有可能造成精度降低或数据溢出，使用时要小心。

## 基本数据类型取值范围大小



## 第三节：常量、变量

### 3.1 常量：常见的数值类型(也称为数据常量)

整型 (int) : 123 int

Java语言整型的三种表示形式：

十进制整数：如12, -314, 0, 8, 7。

八进制整数：要求以0开头，如0127表示八进制的127。

十六进制数：要求0x或0X开头，如0x1df3。

浮点数(实型) : 3.14 float ... 字符: 'a' char 布尔(逻辑): true、false boolean 字符串

String: "a"、"abc"、"hello world", "" null: 表示的引用为空对象

### 3.2 变量：保存程序执行中的数据

变量其实是内存中的一小块区域，使用变量名来访问这块区域。因此，每一个变量使用前必须要先申请(声明)，然后必须进行赋值(填充内容)，才能使用。

### 3.3 变量的声明初始化

变量的声明格式：

1. 数据类型 变量名; 变量名 = 变量值;
2. 数据类型 变量名 = 变量值;

### 3.4 变量的分类

1. 按所属的数据类型划分：

基本类型变量：持有原始值。(原始数据类型)

引用类型变量：持有引用值。(某个对象的引用，不是该对象本身)String 字符串

2. 按被声明的位置划分：

局部变量：方法或语句块内部定义的变量

全局变量：方法外部、类的内部定义的变量

### 3.5 局部变量和全局变量的区别

1. 声明的位置不同：

局部变量的作用域就是它所在的方法或语句块

全局变量的作用域是整个类体。

2. 是否拥有默认值：

全局变量有默认值

局部变量没有默认值

## 第四节：运算符

### 4.1 运算符的分类

1. 算术运算符: + 、 - 、 \* 、 / 、 % 、 ++ 、 --
2. 赋值运算符: = 、 += 、 -= 、 \*= 、 /= 、 % =
3. 关系运算符: > 、 < 、 >= 、 <= 、 == 、 !=
4. 逻辑运算符: ! 、 & 、 | 、 ^ 、 && 、 ||
5. 位运算符: & 、 | 、 ^ 、 ~ 、 >> 、 << 、 >>>
6. 字符串连接运算符: +

## 4.2 算数运算符

```
// + 、 - 、 * 、 / 、 %  
// 需求 : 三位数 求十位上 的数字  
// 123 2  
int a = 123;  
int b = a/10%10;  
// 123/10 12%10 2  
int c = a%100/10;
```

```
// ++ 、 --  
int d = 5;  
System.out.println("d = " + (d++)); // 自身加1  
System.out.println("d = " + (d)); // 自身加1  
  
int e = d++ + 6;  
System.out.println("e = " + e); // 自身加1  
  
// -- 放在前面是先自身减一 在参与运算  
// 放后面 先参与运算 在自身减一  
int f = d-- + 6;  
System.out.println("f = " + f); // 自身加1  
  
byte z = -128;  
byte w = --z; // ++ 在临界值不会报异常 赋的值实在 取值范围内转圈  
System.out.println("w = " + w); // 自身加1
```

## 4.3 赋值运算符

```
int i = 34;
//i = i + 45;
i+=45;
i-=45;

// i = i -45;
i*=45;
i/=45;
i%=45;
System.out.println("i= "+ i);
```

#### 4.4 关系运算符

```
boolean b = 3>=3; // 关系运算符 最终的结果都是boolean类型的
boolean b2 = 3==2; // == 判断等号两边是否相等 如果相等就是true
boolean b3 = 3!=2; // != 判断等号两边是否不等 如果不等就是true

System.out.println(b3);

int a = 4;
int c = 5;
boolean b4 = ++a>6 & ++c>3;
System.out.println(a);
System.out.println(c);
// 短路与 如果左边是 false 右边则不再执行
// 短路或 如果左边是true 右边则不再执行 开发中使用比 & 和 |用的多
```

#### 4.5 逻辑运算符

```

// == 判断基本数据类型的值 是否相同的    != 判断基本数据类型的值是否不等
// 引用数据类型 equals
boolean flag = i!=34;
System.out.println("flag = "+ flag);

boolean flag2 = false&false;// & 必须是左右两边都是 true 结果才是true
boolean a = false | false;// | 只要有一个true 结果就是true 两边都是false 结果才为false

System.out.println("flag2 = "+ flag2);
System.out.println("a = "+ a);

// && || 有可能有一部分是执行不到的

// && 左边如果是 false 就不执行右边 最终结果是false
// || 左边是true 不用执行右边了 最终结果就是 true

int z = 5;
int w = 5;
boolean b = ++z>5 || ++w >5;// w 并没有被执行
System.out.println("b = "+ b+" z="+z+" w="+w);

// ! 非 反
boolean e = !true;
System.out.println("e = "+ e);

// ^ 异或 相同则为false 不同为true
boolean f = true ^ false;
System.out.println("f = "+ f);

```

#### 4.6 字符串连接符

```

System.out.println("Hello"+"World!");
System.out.println("Hello = "+ 5);// 字符串和基本数据类型变量 常量连接以后 都变成了 字符串
System.out.println("Hello = "+ 5+ 8);
System.out.println(5+"");

```

#### 4.7 三目运算符



```

//X ? Y : Z
// 1 X 必须是boolean类型表达式
// 2 Y Z 必须数据类型保持一致

// 执行顺序
// 先执行X      true   执行 Y
//              false  执行 Z

int score = 99;
//boolean falg = score>80;
String str = score>80? "非常优秀" : "优秀";
char c = '男';
int i = c=='男'? 5 : (int)(4.0);
// y 和 z 最好是保持数据类型一致
// 如果不一致 也必须保证 接收的 变量能够存储 y和 z的 数据类型

System.out.println(i);
// 需求: 大于90 非常优秀      大于等于60 及格      小于60 集训努力
String str2 = score>90?"非常优秀":score>=60?"及格":"下个班见";
System.out.println(str2);

```

#### 4.8 表达式

表达式是符合一定语法规则的运算符和操作数的序列

```

0 != **i % 10, a = 0, a==0
5.0 + a
(a - b) * c - 4
i < 30 && i % 10 != 0

```

表达式的类型和值

对表达式中操作数进行运算得到的结果称为表达式的值

#### 第六节：总结

默写：

1. 源文件必须以.java结尾
2. 一个类里可以有main方法(主方法),即程序的入口,程序要执行的部分  
其格式为: `public static void main(String args[]){}`
3. 一个源文件里可以有多个class声明的类,但只能有一个class被public修饰
4. 被public修饰的类的类名必须与源文件名相同
5. 每个语句必须由";"结尾
6. 执行程序分为两步:
  - 1)编译:使用javac命令,将.java的源文件编译成.class字节码文件.  
注意:必须包含扩展名.java
  - 2)运行:使用java命令,虚拟机将字节码文件载入内存,开始执行.  
注意:必须不包含扩展名.class
- 7.java里三种注释的方式:
  - 1)单行注释: `//注释的内容`
  - 2)多行注释: `/*注释的内容,可多行*/`
  - 3)文档注释: `/**文档注释的内容,将被生成进帮助文档中*/`

## 作业

1. 初始化a=3, b=4; 请计算:
  - 1) `(a++)/3+ (--b)*2- (a--)%6+(b++)*3- (b--)`
  - 2) `(++b)*2- (a--)%4+(a++)*5- (--b)/2+ (--a)`
  - 3) `(a--) *6+(b++)/3- (--a)*2- (--b)*2+(++a)`
2. 初始化a=3, b=4; 请计算:  
`a>b&(a++)>b&(--b)<a&(b++)>(a--);`  
`b<=a|(b--)>(--a)|(++b)!=(a++)|(b--)==(a--)`
3. 成绩score=80, 如果成绩大于等于60, 输出及格, 如果成绩大于等于70, 输出中, 如果成绩大于等于80, 输出良, 如果成绩大于等于90, 输出优秀, 如果成绩小于60, 输出不及格。
4. 用户输入四位会员号: 计算四位会员号的之和。提示: 用/和%得到个位, 十位, 百位, 千位

## 面试题

1. 如何不借助第三个变量, 交换两个数据
2. char型变量中能不能存贮一个中文汉字? 为什么?
3. String是最基本的数据类型吗?