

Java 基础语法 回顾：

1 计算机组成：硬件和软件

硬件：输入设备、输出设备、CPU、存储器、其他设备

软件：系统软件、应用软件

2 dos 命令

cd 改变当前目录

cd .. 上一级目录

cd \ 根目录

cd aaa\bbb\ccc

d: 切换到D盘

dir 列出当前目录下的文件和文件夹

/p /s /w /o

md 创建目录

创建一级目录和多级目录

创建aaa

rd 删除目录(空目录)

del 删除一个或一组文件

copy 复制 只能复制文件

xcopy myFile myPhoto /s /e

cls 清屏

exit 退出dos

mspaint 打开画图板

notepad 打开记事本

calc

3 java 三个平台

java SE 标准版 桌面应用程序

java EE 企业版 开发Internet 应用程序：淘宝，京东，移动

java ME 微型版(不用)

4 java 发展

java 之父：詹姆斯 高斯林

jdk 1.0 95

jdk 1.2

jdk 1.5 jdk5.0

jdk 6.0

jdk 7.0

jdk 8.0 2014

jdk 9.0

jdk 10.0

jdk 11.0

5 java 运行核心机制

jvm java 虚拟机 java 跨平台的语言

垃圾回收器

6 jdk 安装 和环境变量的配置

添加 JAVA_HOME

在path 的前面加上 %JAVA_HOME%\bin;

7 第一个hello world

public：公共的

class：表示类

```
HelloWorld:      类名
static :         静态的
void :           无返回值
main:            主函数(方法), 程序的入口
String[] :       字符串数组
args:            参数
system.out.println("hello world");           输出 "hello World"
8 print         : 没有换行
println        : 有换行
printf         : 格式化输出 %f %.2f %d %s %x (X)
9 注释 :
  java 注释有三种: 单行注释、多行注释、文档注释
  //    单行注释
  /* .....
  ..... 多行注释
  .....
  */
```

今天任务

1. 标识符
2. 标识符命名规范和命名习惯
3. 关键字、保留字
4. 常量、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

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. 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（驼峰式命名法）
3. 类名：单词首字母大写，多个单词组成每个单词首字母大写（帕斯卡命名法）

合法的标识符：

HelloWorld、username2、user_name、
_userName、\$abc_123、schoolName、
classNo

非法的标识符：

2UserName、user#Name、Hello World

1.4 关键字、保留字

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：常量、变量

2.1 常量

也称为数据常量，程序运行中固定不变的数据。

整型常量：123、100、200、250、500

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。

浮点数常量(小数)：3.14float ...

字符常量：'a'

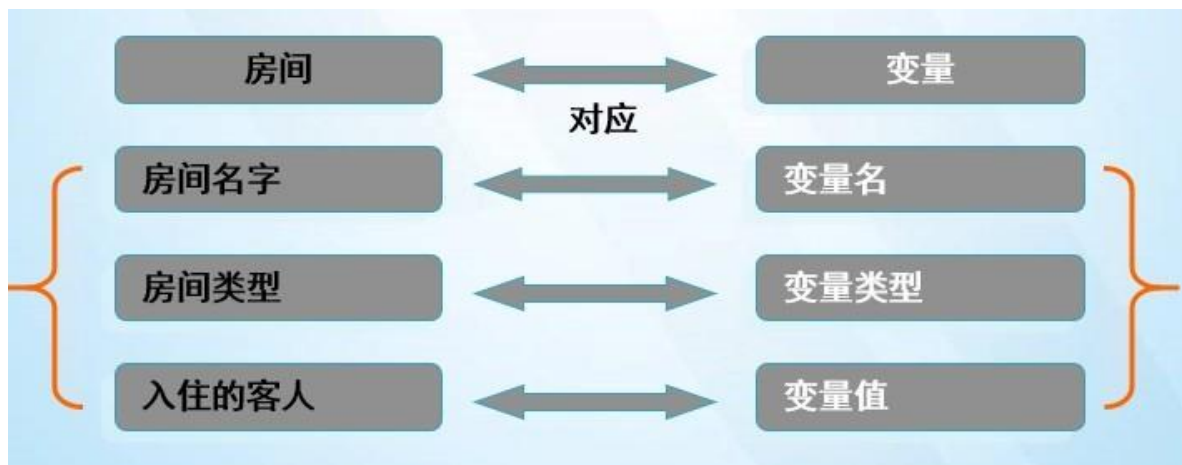
布尔(逻辑)常量：true、false

字符串(String)常量："a"、"abc"、"hello world"、""

null常量：表示的引用为空对象

2.2 变量

变量其实是内存中的一块存储空间，使用变量名来访问这块空间。因此，每一个变量使用前必须要先声明，然后进行赋值(填充内容)，才能使用，变量中存储的数据就是常量。



2.3.1 变量声明、初始化

变量的使用：

1. 先声明，再赋值(初始化)
数据类型 变量名;
变量名 = 变量值;
举例: `int i;`
`i=100;`
`i=200;`
2. 声明同时赋值
数据类型 变量名 = 变量值;
`int j=20;`

上机练习：

练习 1：银行账户存储 1000 元，年利率是 0.05，计算一年后账户余额？ 要求：使用变量保存数据

练习 2：使用变量存储数据，实现个人简历信息的输出 姓名，年龄，工作年限，技术方向、兴趣爱好 要求：使用变量保存数据

2.3.2 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。
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
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
成员变量：方法外部、类的内部定义的变量

2.3.3 局部变量和成员变量

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
成员变量的作用域是整个类体。
2. 是否拥有默认值：
成员变量有默认值
局部变量没有默认值

第三节：数据类型

Java 数据类型可分为两大类:基本数据类型和引用数据类型

基本数据类型: byte、short、int、long、float (单精度)、double (双精度)、char、boolean 引

用数据类型: 类、接口、数组、枚举 (后面讲解)、注解



3.1 基本数据类型

整数型---- byte (字节) , short (短整型) , int (整型) , long (长整型)
浮点型---- float (单精度浮点型) , double (双精度浮点型)
字符型---- char 布尔型---- boolean

基本数据类型的存储长度

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

byte : 1个字节 , 范围 - 128 到 127

short : 2个字节 , 范围 - 32768 到 32767

int : 4个字节

long : 8个字节

float : 4个字节 - 3.40E+38 ~ +3.40E+38 有效位数7-8位

double : 8个字节 - 1.79E+308 ~ +1.79E+308 有效位数15-16位

char : 2个字节 0- 65535 65536 个字符

boolean : 不确定

3.3 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

3.4 浮点型

float : 单精度浮点数 (精度7, 8位) 4个字节

double : 双精度浮点数 (精度15, 16位) 8个字节

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 = 3.14 f ;

注意: **float** **double** 存储的数据都是近似值。

3.5 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");
```

3.6 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 ;
```

3.7 基本数据类型总结

JAVA基本数据类型总结			
NO.	数据类型	大小/位	可表示的数据范围
1	long(长整数)	64	-9223373036854775808~9223372036854775807
2	int(整数)	32	-2147483648~2147483647
3	short(短整数)	16	-32768~32767
4	byte(位)	8	-128~127
5	char(字符)	16	0-65535
6	float(单精度)	32	-3.4E38~3.4E38
7	double(双精度)	64	-1.7E308~1.7E308
8	boolean	不确定	true 、 false

3.8 基本数据类型的转换

1 基本数据类型之间的转换分为以下两种:

- * 自动类型转换

- * 强制类型转换

特殊: `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`

- * `byte`, `short`, `char` 三者 在计算时会转换成 `int` 类型 (了解)

- * `int` 整型常量和字符常量在合理范围内可以直接赋值给 `byte`、`short`、`int`、`char` (了解)

上机练习¹:

某班第一次Java考试平均分81.29, 第二次比第一次多2分, 计算第二次考试平均分。

- * 如果一个操作数为 `double` 型, 则整个表达式可提升为 `double` 型。

上机练习²:

```
byte b1=10;
```

```
byte b2=20;
```

```
byte b3=b1+b2; // 有问题吗?
```

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。

上机练习:

去年Java所占市场份额是20, 今年增长的市场份额是9.8, 求今年所占份额?

```
int beforeyear =20;
```

```
int year =(int)( beforeyear +9.8 );
```



第四节：运算符和表达式

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：&、|、^、~、>>、<<、>>>（了解）
- 6. 字符串连接运算符：+
- 7. 三目运算符：?

4.2 算数运算符：

描述	运算符	运算符	描述
加	+	%	求余、取模
减	-	++	自增
乘	*	--	自减
除	/		

```
上机练习1： 有一个三位数，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;
```

```
上机练习2：
int d = 5;
System.out.println (( d++)); // 自身加1
System.out.println (( 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，赋的值是在取值范围内转圈
System.out.println ("w=" + w); // 自身加1
```

4.3 赋值运算符：

运算符	描述	运算符	描述
=	赋值 比如:int i=10;	/=	除等于
+=	加等于	%=	模等于
-=	减等于		

<code>*=</code>	乘等于		
-----------------	-----	--	--

```

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

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

```

上机练习

•要求：有一个四位数字的会员卡号，使用 “/” 和 “%” 运算符分解获得会员卡各个位上的数字，将各个位上数字求和

4.4 关系运算符：

运算符	描述	运算符	描述
<code>==</code>	相等于	<code><=</code>	小于等于
<code>!=</code>	不等于	<code>>=</code>	大于等于
<code><</code>	小于		
<code>></code>	大于		

```

boolean b = 3>=3; // 关系运算符 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);

```

4.5 逻辑运算符： 逻辑运算符用于对 boolean 型数据进行运算，运算结果总是 boolean 型

运算符	描述	运算符	描述
<code>&</code>	逻辑与	<code>&&</code>	短路与
<code> </code>	逻辑或	<code> </code>	短路或
<code>^</code>	异或		
<code>!</code>	逻辑非		

```

public static void main(String[] args){
    //逻辑与 或
    System.out.println(true&true);
    System.out.println(true&false);
    System.out.println(true|true);
    System.out.println(true|false);

    System.out.println(!true);
    //异或 : 不相同为真, 相同为假
    System.out.println("-----");
    System.out.println(true^false);
    System.out.println(true^true);
    System.out.println(false^false);
    // && 短路 与
    System.out.println("*****");
    int num1=10;
    int num2=5;
    boolean b=num1<num2 && num1++>100; //num1<num2 false
    System.out.println(b);
    System.out.println(num1);
    //|| 短路或
    boolean b2=num1>num2 || num1++>100;
    System.out.println(b2);

    System.out.println(num1);
}

```

4.6 位运算符(了解)

运算符	描述	运算符	描述
&	按位与	>>	右移
	按位或	<<	左移
^	按位异或	>>>	无符号右移
~	按位取反		

4.7 字符串连接符 + 如果字符串和基本数据类型变量、常量连接以后，结果变成了字符串。

```

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+"");

```

4.8 三目运算符 ? :

X ? Y : Z

X 为 boolean 类型表达式，先计算 x 的值，若为 true，整个三目运算的结果为表达式 y 的值，否则整个运算结果为表达式 z 的值。示例：

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

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 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输出"非常优秀", 大于等于60"输出及格", 否则小于60输出"下个班见"
String str2 = score >90?"非常优秀": score >=60?"及格": "下个班见";
System.out.println (str2 );
```

4.9 运算符的优先级

运算符	结合性
[] . () (方法调用)	从左向右
! ~ ++ -- -(负号)	从右向左
* / %	从左向右
+ -	从左向右
< < > > >	从左向右
< <= > >= instanceof	从左向右
== !=	从左向右
&	从左向右
^	从左向右
	从左向右
&&	从左向右
	从左向右
?:	从右向左
=	从右向左

4.10 表达式

表达式：符合一定语法规则的运算符和操作数的序列

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

表达式的值和类型

* 对表达式中操作数进行运算得到的结果称为表达式的值

* 表达式的值的数据类型即为表达式的类型

第五节：Scanner 简单使用

//Scanner 的作用：实现键盘输入数据。

//Scanner 的使用

//1 先导入Scanner import java.util.Scanner;

//2 创建对象 Scanner input=new Scanner(System.in);

//3 接收键盘录入的数据

```
import java . util . Scanner ;
```

```
public class Demo1{
```

```
    public static void main( String [] args ){
```

```
        Scanner input =new Scanner ( System . in );
```

```
        //3.1 接收字符串类型数据next(), 遇到空格截断nextLine(); 遇到换行结束
```

```
        //String s=input.next();
```

```
        String s=input . nextLine () ;
```

```
        System . out . println ( s);
```

```
        //3.2 接收整型数据nextInt();
```

```
        int num=input . nextInt () ;
```

```
        System . out . println ( num);
```

```
        //3.3 接收浮点数
```

```
        double d=input . nextDouble () ;
```

```
        System . out . println ( d);
```

```
    }
```

```
}
```

需求：从控制台输入学员 3 门课程成绩(html, java, sql)，编写程序实现

(1) Java 课和 SQL 课的分差之差

(2) 3 门课的平均分

```
import java.util.Scanner; //导入 Scanner 类 public class
```

```
ScoreStat {
```

```
    public static void main(String[] args) {
```

```
        //创建对象
```

```
        Scanner input = new Scanner(System.in);
```

```
        System.out.print("htmls 的成绩是: ");    int html =
```

```
        input.nextInt(); //html 分数    int java = input.nextInt();
```

```
int sql = input.nextInt();    int diffen;    //分数差
double avg;    //平均分

//省略输出成绩单代码.....    diffen = java - sql; //计算 Java 课和
SQL 课的成绩差    System.out.println("Java 和 SQL 的成绩差: " +
diffen);    avg = (html + java + sql) / 3; //计算平均分
System.out.println("3 门课的平均分是: " + avg);    }}
```

总结

1 标识符 给包、类、方法名、变量、参数定义的一个字符序列。

2 标识符命名规范

1> 只能包括字母、数字、下划线、\$

2> 不能以数字开头

3> 区分大小写

4> 长度无限制(不要超过15)

5> 不能是java 关键字

3 标识符命名习惯:

1> 见名知意

2>方法名、变量名、参数名首字母小写，后面首字母大写, 驼峰式命名法

3>类名采用首字母大写，帕斯卡命名法

4 关键字和保留字

public class static

void 等等 5 常量和变

量 常量：固定的数

据。

整型常量:10,20 浮点常量:3.14 字符

常量: 'a' 布尔常量:true false 字符串常

量: "xxxx","hello world" null 常量

变量：本质就是内存中一块存储空间。

变量名，变量的名字 变量

值，变量中存储的数据 变

量类型 使用变量：

1》先声明，再赋值

```
int num;
```

```
num=20;
```

2>声明的同时，赋值

```
int num2=20;
```

变量分类

1 按照类型 分： 基本数据类型变量和引用类型变量

2 按照声明的位置分： 局部变量和成员变量

6 数据类型两大类：基本数据类型和引用数据类型

基本数据类型 : byte short int long float double char boolean

引用类型 : 类 String 、数组 、接口、 枚举

byte : 1 个字节 -128-127

short : 2 -32768-32767 int : 4

long : 8 float: 4 double : 8

char : 2 0-65535 boolean : 不

确定

基本数据类型之间的转换

1 自动类型转换 : 表示范围小的转成表示范围大的

byte---->short---->int---->long --->float--->double char--

->int byte short char 运算是转成 int

int 常量和字符常量在合理的范围之内可赋值给 byte short int char

2 强制类型转换 表示范围大的转成表示范围小的

7 运算符

算术运算符: + - * / % ++ -- 赋值运算符: = +=

-= *= /= %=

关系运算符: == != > < >= <=

逻辑运算符: & | ! ^ && ||

位运算符: & | ~ ^ >> << >>> (了解)

字符串连接符 : + 三目运算

符 : X ? Y : Z

优先级 表达

式

8 Scanner 的使用

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. `jav` 注释:
 - 1) 单行注释 `// 注释的内容`
 - 2) 多行注释 `/* 注释的内容 可多行*/`

作业

1.初始化 `a=3, b=4`; 请计算:

1) `(a++)/3+ (--b)*2-(a--)%6+(b++)*3-(b--)` `1+6-4+9-4=8`

2) `(++b)*2-(a--)%4+(a++)*5- (--b)/2+ (--a)` `10-3+10-2+2= 17`

3) `(a--)*6+(b++)/3- (--a)*2- (--b)*2+(++a)` `18+1-2-8+2=11`

2.初始化 `a=3,b=4`;请计算:

`a>b&(a++)>b&(--b)<a&(b++)>(a--);` `false&false&true&false =false` `b<=a|(b--)>(--`

`a)|((++b)!=(a++))|(b--)==(a--)` `false | true |true | false =true`

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` 型变量中能不能存储一个中文汉字? 为什么?
3. `String` 属于基本数据类型吗?