# Day02

## 关键字

|  |
| --- |
| 定义：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）。  **特点：全部为小写** |

## 标示符

### 标示符概念

|  |
| --- |
| 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。有了标示符，就可以对这些元素进行访问和操作。 //goto |

### 标示符命名规则(法律)

|  |
| --- |
| 1、由字母、数字、下划线（\_）和美元符号（$）组成。  2、不能以数字开头。  3、区分大小写。A 和a 是两个完全不同的标示符。  4、长度无限制。（习惯，不要超过15个字符）  5、**不能是Java中的保留字和关键字。**  **注意：在写标示符的时候尽量有意义！！！**  **规则必须要执行！！！** |

### JAVA中命名规范

|  |
| --- |
| 1、包名：多单词组成时所有字母都小写。  xxxyyyzzz  2、类名接口名：多单词组成时，所有单词的首字母大写。  XxxYyyZzz  3、变量名和函数名：多单词组成时，第一个单词首字母小写，第二个单词开始每个单词首字母大写。  xxxYyyZzz  4、常量名：所有字母都大写。多单词时每个单词用下划线连接。  XXX\_YYY\_ZZZ |

## 常量

|  |
| --- |
| 常量表示不能改变的数值。  Java中常量的分类：   1. 整数常量。所有整数 2. 小数常量（浮点数）。所有小数 3. 布尔型常量。较为特有，只有两个数值。true false。 4. 字符常量。将一个数字字母或者符号用单引号( ' ' )标识。 ‘ab’ 5. 字符串常量。将一个或者多个字符用双引号标识。 “fasdfla” 6. null常量。只有一个数值就是:null.   对于整数：java有4种表现形式：  二进制：0-1 满2进1 用0b开头表示**(java 7新增支持)**  十进制：0-9 ，满10进1.  八进制：0-7 ，满8进1. 用0开头表示。  十六进制：0-9，A-F，满16进1. 用0x开头表示。  注意：在写数值的时候数值太长的时候，会影响可读性，**java7新增数值的下划线支持**  。例如：124559656566 可以根据需要任意加下划线，但是不能加在开始和结束。124\_559\_656\_566 来增加可读性。 **\_12错误！！！ 24\_ 错误！！！** |

## 变量

|  |
| --- |
| **变量：**程序执行中数值可变的数据  **变量名：**定义变量的标识符  **变量值：**内存单元中所装载的数据  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)(强类型语言的特点)，然后必须进行赋值(填充内容)，才能使用。  如：  int i = 100;  int y;  y = i +10; |

## JAVA数据类型

|  |
| --- |
|  |
| Java中定义了四类/八种基本数据类型(**必须记住**)  布尔型---- boolean  字符型---- char  整数型---- byte, short, int, long  浮点数型---- float, double  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。 |

|  |
| --- |
| 计算机中的数据都以二进制数字保存。  二进制：逢二进一。即只有0、1两个值。  如：十进制的10在计算机内保存为二进制的1010  计算机中信息的存储单位  位（Bit）：表示一个二进制数码0或1，是计算机存储处理信息的最基本的单位。  字节（Byte）：一个字节由8个位组成。它表示作为一个完整处理单位的8个二进制数码。 |

### 整数类型

|  |
| --- |
| Java语言的整型常量默认为int型，如：int i = 3;  要声明long型可以加‘l’或‘ L’，如：long l = 3L; |

### 浮点数据类型

|  |
| --- |
| 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; |

### char数据类型

|  |
| --- |
| char型数据用来表示通常意义上“字符”  char c = 'A';  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';  Char c2=97;//ASCII美国标准信息交换码  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'; |

|  |
| --- |
|  |

|  |
| --- |
| ASCII美国标准信息交换码：计算机中用得最广泛的字符集及其编码,由美国国家标准局(ANSI)制定.它已被国际标准化组织（ISO）定为国际标准，称为ISO 646标准。适用于所有拉丁文字字母，ASCII码有7位码和8位码两种形式。  因为1位二进制数可以表示（21=）2种状态：0、1；而2位二进制数可以表示（22）=4种状态：00、01、10、11；依次类推，7位二进制数可以表示（27=）128种状态，每种状态都唯一地编为一个7位的二进制码，对应一个字符（或控制码），这些码可以排列成一个十进制序号0～127。所以，7位ASCII码是用七位二进制数进行编码的，可以表示128个字符。   第0～32号及第127号(共34个)是控制字符或通讯专用字符，如控制符：LF（换行）、CR（回车）、FF（换页）、DEL（删除）、BEL（振铃）等；  通讯专用字符：SOH（文头）、EOT（文尾）、ACK（确认）等；   第33～126号(共94个)是字符，其中第48～57号为0～9十个阿拉伯数字；65～90号为26个大写英文字母，97～122号为26个小写英文字母，其余为一些标点符号、运算符号等。   注意：在计算机的存储单元中，一个ASCII码值占一个字节(8个二进制位)，其最高位(b7)用作奇偶校验位。所谓奇偶校验，是指在代码传送过程中用来检验是否出现错误的一种方法，一般分奇校验和偶校验两种。奇校验规定：正确的代码一个字节中1的个数必须是奇数，若非奇数，则在最高位b7添1；偶校验规定：正确的代码一个字节中1的个数必须是偶数，若非偶数，则在最高位b7添1。 |

### boolean数据类型

|  |
| --- |
| 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(c语言可以)。  示例：  boolean b = false; |

### java八大基本数据类型总结

|  |
| --- |
|  |

### 基本类型之间的数据转换

|  |
| --- |
| **boolean 类型不能转换成任何其它数据类型。**  自动类型转换：容量小的类型自动转换成容量大的数据类型  byte,short,intlongfloatdouble (范围依次变大)  char  byte和char与short和char之间不会互相转换，它们三者在计算时会转换成int类型  强制类型转换：容量大的类型转换成容量小的数据类型时，要加上强制转换符  long l = 100L;  int i = (int)l;  有可能造成精度降低或数据溢出，使用时要小心。 |

## 运算符

|  |
| --- |
| 算术运算符：+ 、 - 、 \* 、 / 、 % 、 ++ 、 --  赋值运算符：= 、 (+= 、 -= 、 \*= 、 /= 、 %=) 复合赋值运算符  关系(比较)算符：> 、 < 、 >= 、 <= 、 == 、 !=  逻辑运算符：! 、 & 、 | 、 ^ 、 && 、 ||  位运算符：& 、 | 、 ^ 、 ~ 、 >> 、 << 、 >>>  字符串连接符：+ |

### 算数运算符

|  |
| --- |
|  |

### 赋值运算符

|  |
| --- |
|  |

### 关系运算符

|  |
| --- |
| 关系运算符作用是比较两边的操作数，结果总是boolean型的。 |

### 逻辑运算符

|  |
| --- |
| 逻辑运算符用于对boolean型结果的表达式进行运算，运算结果总是boolean型。 |

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **操作数** | **操作数** | **&** | **|** | **&&** | **||** |  | **^** |
| true | false | f | t | f | t |  | t |
| true | true | t | t | t | t |  | f |
| false | false | f | f | f | f |  | f |
| false | true | f | t | f | t |  | t |

### 位运算符

|  |
| --- |
| 位运算符对两个操作数中的每一个**二进制位**都进行运算  位运算符功能  按位取反 ~  按位与 &  按位或 |  按位异或 ^ |

### 移位运算符

|  |
| --- |
| 1. 左移："a<<b; "将二进制形式的a逐位左移b位，最低位空出的b位补0 2. 带符号右移："a>>b; "将二进制形式的a逐位右移b位，最高位空出的b位补原来的符号位 3. 无符号右移："a>>>b;"将二进制形式的a逐位右移b位，最高位空出的b位补0 |

### 字符串连接运算符

|  |
| --- |
| String s="He" + "llo"; 结果"Hello"  "+"除了可用于字符串相连接，也能将字符串与其它的数据类型相连成一个新的字符串。  如：String s="x" + 123; 结果"x123” |

### 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。  例：  int score = 75;  String type = score >=60 ? "及格" : "不及格"; |

## 表达式

|  |
| --- |
| 定义：表达式是符合一定语法规则的运算符和操作数的序列  a = 0  5.0 + a  (a - b) \* c - 4  i < 30 && i % 10 != 0  1、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  表达式的值的数据类型即为表达式的类型  2、表达式的运算顺序  首先应按照运算符的优先级从高到低的顺序进行  优先级相同的运算符按照事先约定的结合方向进行 |