# day01

## Java环境检查

1. win+R打开运行窗口输入cmd，按Enter打开dos命令窗口，然后输入“java -version”命令按Enter，如果出现"Java不是内部或外部的命令"，则说明没有安装JDK，如果出现如图所示的内容则说明安装过JDK，JDK的版本为1.8.0

```
C:\Users\Lenovo>java -version
java version "1.8.0_101"
Java(TM) SE Runtime Environment (build 1.8.0_101-b13)
Java HotSpot(TM) 64-Bit Server VM (build 25.101-b13, mixed mode)

C:\Users\Lenovo>
```

2. 在dos命令窗口中输入"javac"命令，如果 出现"javac不是内部或外部的命令"，则说明没有配置环境变量，若果出现如图所示则说明配置过环境变量

```
C:\Users\Lenovo>javac
用法: javac <options> <source files>
其中, 可能的选项包括:
  -g                         生成所有调试信息
  -g:none                    不生成任何调试信息
  -g:{lines,vars,source}     只生成某些调试信息
  -nowarn                    不生成任何警告
```

3. 配置环境变量步骤

   3.1.打开计算机属性，选择高级系统设置，选择高级下面的环境变量

   3.2.在用户变量或者系统变量中设置

   3.2.1.若将环境配置在用户变量中则只对当前所登录的用户有效

   3.2.2.若将环境配置在系统变量中则对当前系统所有用户都有效

   3.3.新建JAVA_HOME变量，变量值设为JDK的安装路径，例如

JAVA_HOME: C:\Program Files\Java\jdk1.8.0_101

3.4.找是否有Path变量，若没有则新建，若有则编辑

3.4.1.若是新建，则之间在变量值中输入%JAVA_HOME%\bin

3.4.2.若是编辑，则将光标移到变量值的最后，检查是否有分号，若没有则添加上分号(英文半角输入法下的符号)，然后再输入%JAVA_HOME%\bin

3.5.最后点击确定

# Java的运行过程

java在运行过程中分为两个阶段：编译期和运行期

编译期：将.java源文件编译成.class字节码文件

运行期：由java虚拟机执行.class字节码文件

名词解释(*):

JVM: java虚拟机

加载并运行.class字节码文件

JRE: java的运行环境

除了JVM以外还包含运行java程序所必须的环境

JDK: java的开发工具包

除了包含JRE以外还包含开发java程序所必须的工具包

运行java程序的最小环境为JRE

开发java程序的最小环境为JDK

# Java的第一段代码HelloWorld

创建java项目:

1.右键New --》 project --》 Java Project ----------小区

2.在src上右键New --》Package ----------单元楼

3.在包名上右键New --》 Class ----------门牌号

注意:Class的名字通常情况下建议首字母大写，如果是多个单词组合而成的则建议每个单词的首字母都大写

```java
public class HelloWorld {
    //单行注释   main方法就是Java主程序的入口
    public static void main(String[] args){
        //输出语句  println 输出并换行
        System.out.println("欢迎来到Java世界");
        // print 输出不换行
        System.out.print("欢迎来到软帝");
        System.out.println("周哥超级帅");

    }

}
```

# Java中的变量

## 变量的声明与初始化

```java
public static void main(String[] args){
        //变量的初始化
        //1.先声明后初始化
        //声明变量的语法格式      数据类型 变量名
        int a;//声明了一个整数类型的变量，名为a;
        //初始化就是给变量第一次赋值，需要注意:初始化的值的数据类型应该与声明的时候的数据类型匹配
        a = 10;//将整数10赋值给变量a
        //2.声明的同时初始化
        int b = 20;//声明了一个整数类型的变量，名为b，并且初始化为20
        //3.同时声明多个相同类型的变量
        int c,d,e;//声明了三个整数类型的变量，分别为c,d,e
        int x,y,z = 10;//声明了三个整数类型的变量，分别为x,y,z，但是只有z初始化为10
    }
```

## 变量的命名规范

1. Java中变量名在同一作用域中不能重复
2. Java中严格区分大小写
3. Java中变量名可以使用中文，但不建议使用
4. Java中变量名尽可能的见名知义
5. Java中变量名可以使用汉语拼音
6. Java中变量名除了$和_以外不能使用任何特殊的符号
7. Java中变量名不能使用数字开头
8. Java中变量名不能使用Java关键字
9. Java中变量名遵循驼峰命名法

```java
public static void main(String[] args){
    //变量的命名规范
    //1.在同一作用域中变量名不能重复
    //      int a;//编译错误
    //2.Java中严格区分大小写
```

```
    int A;
    //3.Java中变量名尽可能的做到见名知义
    int age;
    //4.Java中变量名可以使用中文命名(不建议使用)
    int 年龄;
    //5.Java中变量名可以使用汉语拼音
    int nianling;
    //6.Java中变量名除了$和_以外不能使用任何特殊的符号
    //      int a#ge;//编译错误
    int $age;
    int ￥age;//使用￥虽然不会编译错误，但是一般不会这样去用，因为在编程过程中所有的符号都是英文半角输入法的符号
    //7.Java中变量名不能使用数字开头
    //      int 2b;//编译错误
    int b2;
    int o2o;
    //8.Java中变量名不能使用Java关键字
    //      int public;//编译错误
    //      int null;//编译错误
    int NULL;
    //9.Java中变量名建议使用驼峰命名法(多个单词组合而成时，第一个单词的首字母小写，后面每一个单词的首字母大写)
    int classNumber;
}
```



## 变量的使用

使用变量就等同于使用了变量中的值

注意: 变量赋值的时候，要注意变量的值要和声明的变量数据类型一致

# Java中的数据类型

- 基本数据类型(简单数据类型)
    1. 整数类型: byte，short，int，long

       byte：1个字节，8位，-128~127 最大存储量为255

       short：2个字节，16位，-32768~32767 最大存储量为65535

       int：4个字节，32位，-2^31~2^31-1 最大存储量为2^31-1

       long：8个字节，64位，-2^64~2^64-1 最大存储了为2^64-1
    2. 浮点数类型(小数类型)：float，double

       float：4个字节，32位

       double：8个字节，64位
    3. 字符型：char

       char：2个字节，16位，存储Unicode编码，用' '
    4. 布尔型：boolean 取值只能是true或者false
- 引用数据类型

除了基本数据类型以外几乎全部都是引用数据类型

**特别注意:String字符串类型，虽然我们用的比较多比较频繁，但是它是引用数据类型**

```
/**
 * 演示Java中的基本数据类型
 * @author Mr.Zhou
 * @date: 2019年11月18日 下午2:46:09
 */
public class DataType {
    public static void main(String[] args){
        //byte 1个字节  -128~127
        byte b;//声明了一个byte类型的变量，名为b
//      b = -129;//编译错误，内存溢出
        b = -128;
        //short 2个字节   -32768~32767
        short s;
//      s = -32769;//编译错误，内存溢出
        s = -32768;
        s = 32767;
        //int 4   -2^31 ~ 2^31-1
        int i = 2100000000;
//      i = 2200000000;//编译错误，内存溢出
//      System.out.println(2100000000+100000000);
        //long 8个字节   -2^64 ~ 2^64-1
        //注意:在初始化long类的变量时，在最后要加上一个L或者l，通常建议添加L
        long lo = 2200000000L;

        //double 8个字节   双精度浮点型
        double dou = 3.14;
        //float 4个字节  单精度浮点型(在初始化float类型的变量时，最后需要加上一个F或者f)
        float f = 3.14F;

        //char 2个字节  字符型
        char ch = '男';//只能存储一个字符
//      char c = '男人';//编译错误

        //boolean 取值只能是true或者false
        boolean boo = true;
        boolean bo = false;


    }

}
```



# 数据类型间的转换

数据类型之间的转换仅存在于数值类型之间，数值类型间的大小关系从小到大依次为

byte，short，int，long，float，double

转换规则: 从小转到大，自动类型转换

从大转到小，强制类型转换

注意:在强制类型转换的过程中可能会造成数据的精度丢失，当浮点数强转为整数时，都是无条件的舍弃掉小数位;在强制类型转换过程中可能会造成数据溢出，所以强制类型转换需要慎重

```
/**
 * 演示Java中基本数据类型之间的转换
 * @author Mr.Zhou
 * @date: 2019年11月18日 下午3:54:42
 */
public class DataChange {
    public static void main(String[] args){
        //byte short int long float double
        short s = 10;
        int i = s;//自动类型转换，short会自动转为int
        short s1 = (short)i;//强制类型转换
        double dou = 3.94;
        int i2 = (int)dou;//在强制类型转换的过程中可能会造成数据的精度丢失，当浮点数强转为整数时，都是无条件的舍弃掉小数位
        System.out.println(i2);

        short s3 = -200;
        byte b = (byte)s3;//在强制类型转换的过程中可能会造成数据溢出，强转需谨慎
        System.out.println(b);
    }

}
```



# Java中的运算符

1. 算数运算符

   + - * / % ++ --

   注意:byte short char在参与算数运算时都会自动转为int类型后参与运算

   Java中两个int类型的数相除，得到的结果一定是一个int类型，而且是小数位无条件舍弃的整数

   若要的到浮点数，则需要将其中一个int类型的整数转为double类型

   % 取余/取模 模运算

   ++/-- 自增/自减

   ```
   /**
    * 演示Java中的运算符
    * @author Mr.Zhou
    * @date: 2019年11月18日 下午4:27:06
    */
   public class Operator {
       public static void main(String[] args){
           //算数运算
           short s1 = 10;
   ```

```
        int i1 = 15;
        double d1 = 3.14;
        int i2 = s1 + i1;
        System.out.println(i2);
        //注意:byte short char在参与算数运算时都会自动转为int类型后参与运算
        short i3 = (short)(s1 + 15);
        char c1 = 'a';//97
        char c2 = 'A';//65
        char c3 = '0';//48
        int i4 = c3 + 1;
        System.out.println(i4);

        int i6 = 11;
        int i7 = 2;
        double i5 = i6/i7;//Java中两个int类型的数相除，得到的结果一定是一个int类型，而且是小数位无条件舍弃的整数
        System.out.println(i5);
        //若要的到浮点数，则需要将其中一个int类型的整数转为double类型
        i5 = (double)i6/i7;
        System.out.println(i5);

        //% 模运算     取余|取模
        int i8 = i6%i7;//i6/i7取余数
        System.out.println(i8);//1

        // ++/--  自增1/自减1
        // ++在前和++在后的区别
        /* 单独使用时，++在前和在后没有任何区别
         * 但是参与运算时，++在前则是先自加1再运算，++在后时则是先运算后自加
         * --同理
         */
        int a = 10;
//      a++;//==>a = a+1
        ++a;
        System.out.println(a);

        int b = 10;
        int c = b++ + 5 + ++b + 2;
        System.out.println(c);//c=29
        System.out.println(b);//b=12


        int x = 5;
        int y = --x + 5 - x-- + ++x + 2 - x++;
        System.out.println(y);//7
        System.out.println(x);//5

    }

}
```

2. 关系运算符
3. 逻辑运算符
4. 赋值运算符
5. 拼接运算符
6. 条件运算符(三目运算符/三元运算符)