**一：简单的命令**

我们先从javac命令开始，javac命令是将源代码编程成class字节码文件，因为我们的JVM虚拟机是执行class字节码文件的，不是执行源代码，JVM虚拟机是不认识源代码的。

**public** **class** **Helloworld{**

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

System**.**out**.**println**(**"hello world"**);**

**}**

**}**

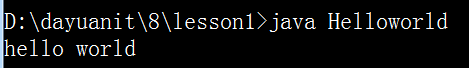
上面是一段简单的java源代码，我们需要使用javac命令将其编程成字节码文件，才能运行，如下图：



执行之后，当前目录下会多出一个Helloworld.class文件，这就是字节码文件，如下图：



如果想运行，那么可以使用java命令来执行class字节码文件：



执行结束后，在控制台，输出hello world字符串。另外，需要注意，并不是每个字节码都能得到执行，只有含有main方法的class文件才能执行，因为，main方法是java程序的入口。

**二：复杂的命令**

当我们的java源文件有包结构的时候，再使用javac命令编译，就需要一些参数。举个例子，我有两个类，一个是Dog类，一个是Test测试类：

package com.dayuanit.animal;

public class Dog {

public void move() {

System.out.println("dog move");

}

}

package com.dayuanit.test;

import com.dayuanit.animal.Dog;

public class Test {

public static void main(String[] args) {

Dog dog = new Dog();

dog.move();

}

}

这两个类，使用javac命令编译如下：

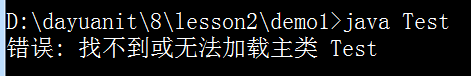


我来解释下上面的命令，这个javac编译比之前的多了一个“-d”参数，这个参数是用来生成包结构的，也就是源代码声明的package目录结构，我们需要将生成的class字节码文件放在这个package下面，通过javac命令，能够帮助我们自动生成package目录，并且能够将class字节码文件放在对应目录的下面，很方便吧：



编译过之后，会在当前目录下生成包目录，如上图，另外，注意，"-d"参数后面有个"."，这个点代表当前目录，当然你也可以改为其他目录。

源代码已经编译好了，那么我们该如果运行呢？



如上图，我们使用java命令直接运行Test，发现报错了。因为此时Test有了包结构，它的全名称应该叫：com.dayuanit.test.Test，所以，我们的命令应该是：



如上图，我们就能够正确运行Test类了，这里强调一下，当类有了包名后，类的名字就是包名+类名了，这才是它的真实名称。

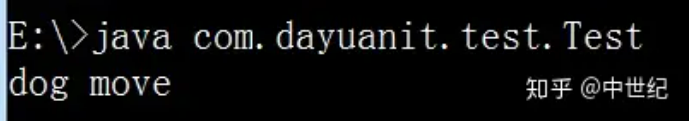
**三：把编译后的字节码放在其他位置。**

我们知道javac命令的-d参数能够帮助我们生成包目录结构，另外，还能指定生成包目录放在哪里，好，我们来看下如何使用。



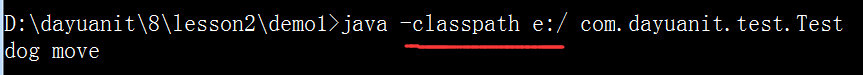
如上图，-d参数后面我跟上 e:/，这代表我将编译后的包目录放在了e盘下面了：

好，我们将字节码放在了e盘，如何运行字节码呢？方法有两个，且看第一种方式：



我们可以到E盘下面，然后运行java命令即可，如上图，因为我们的包在E盘下面。

第二种方式：



如上图，这次我们还是在源代码目录，编译之后，字节码都在E盘下面，但是我们要在源代码目录下直接运行E盘下面的字节码，我们在java命令后面跟上了 -classpath参数，这个参数是告诉java虚拟机，到E盘下面找我们的class字节码文件，当然你也可以指定多个目录，用“；”分号隔开即可。