

四、我编的第一个程序

```
package com.ch01;

public class HelloWorld
{
    public static void main(String args[])
    {
        System.out.println("Hello World!");
    }
}

javac -d . ch01/HelloWorld.java
javac -d . ch01/*.java
运行:
java com.ch01.HelloWorld
```

1、package 包（1）为.class 文件归类

（2）有效的避免命名冲突

（3）一个类如果定义了包，在编译之后，类名发生改变

识别格式：包名.类名 类的全名

2、javac 将 java 源文件编译成.class 文件

—d 指定.class 所存放的目录

五、基本语法

1

```
if(boolean) {}
```

2

```
if(boolean)
```

```
{
```

```
}
```

```
else
```

```
{
```

```
}
```

3

```
if(boolean)
```

```
{
```

```
}
```

```
elseif(boolean)
```

```
{
```

```
}  
else  
{  
  
}
```

Loop

1、
for((1);(2);(3))
{
 (4)
}

- (1) 为控制循环的变量进行初始化
- (2) 判断循环是否执行的条件
- (3) 每一次循环之后对控制循环的变量做计算
- (4) 循环体

2、
(1)
while ((2))
{
 (4)
 (3)
}

3、
(1)
do
{
 (4)
 (3)
} **while**(2);

我编的第二个程序

```
package com.ch01;  
  
public class LoopTest  
{  
    public static void main(String args[])  
    {
```

```

        int i=0;
        int sum=0;
        //(1)for:
        for(i=1;i<=100;i++)
        {
            sum+=i;
        }
        System.out.println("for-sum "+sum);

        //(2)while:
        i=1;
        sum=0;
        while(i <= 100)
        {
            sum+=i;
            i++;
        }
        System.out.println("for-sum "+sum);

        //(3)do-while
        i=1;
        sum=0;
        do
        {
            sum+=i;
            i++;
        }while(i <= 100);
        System.out.println("for-sum "+sum);
    }
}

break 跳出整个循环
continue 结束本次循环, 重新考试下一次循环

```

六、数组

- 1、概念：一组（数据类型相同）的数据的（有序）的（集合）
一组地址号连续的内存节点
- 2、何时使用数组，何时使用链表？
 - (1) 链表开辟内存的成功率相对较大
 - (2) 链表在做插入和删除操作的时候，效率相对较高
数组可能在上述操作中发生整批的数据 **copy**，效率相对较低
 - (3) 查询：数组检索效率相对较高
- 3、语法

```
int [] array=new int[3];  
int array []=new int[3];  
int *p
```

我编的第三个程序

```
package com.ch01;  
  
public class ArrayTest  
{  
    public static void main(String args[])  
    {  
        int array []=new int[100];  
        int sum=0;  
  
        for(int i=0;i<100;i++)  
        {  
            array[i]=i+1;  
            sum+=array[i];  
            System.out.print(array[i]+" ");  
        }  
        System.out.println();  
        System.out.println("sum "+sum);  
    }  
}
```