# 环境变量的配置

# 第一个程序

public class FirstProgram{

public static void main(String[] args){

System.out.println(“Hello world!”)

# 系统输出语句

# 变量

* **常量：**

1. 在程序运行过程中，常量值 是不变的；
2. 加final修饰；让其内容不能重新赋值；
3. 变量名，（约定的）全部大写；

为什么要有变量？-🡪用来计算-🡪数据、字符/字符串、布尔值;

存储计算后的结果数据（中间数据）。

人脑：分为计算区域和存储区域；

电脑：CPU、内存

* 人与人交流：自然的语言
* 人与机交流：程序语言
* 计算机内部：计算机语言（0、1）

变量的有效范围（有效期）：也就是作用域（在**{ }**里有效）；

* **变量**：做计算用来临时存储数据的；

在**内存**中开辟出的一个空间（区域），用来**暂时存储**数据。

“**变**”量：该区域可以重复存放不同数据；

数据类型 变量名; --🡪申明变量；

变量名 = 对应类型的数据; --🡪赋值语句；

变量 （表达式）：

* **变量数据类型**

**short int long char String boolean byte double**

1. **自动类型转换：低精度向高精度转换；**
2. **强制类型转换：高精度向低精度转换；**

Java的整型常量默认为**int**;

浮点型常量默认为**double**;

声明float型常量：

float f=10f/10F;

a = (数据类型)b

* **变量命名规范**：

1. 不能使用关键字；
2. 首字母不能是数字;
3. 可使用的字符范围：字符，数字，下划线，$；
4. 不能有空格；
5. 见名知义；
6. 建议采用**驼峰式**写法（首字母小写，之后的单词首字母大写）；

# 运算符

1. 算术运算符

**+ - \* / %（取余/取模）**

/： 被除数和除数只要有一个数是浮点数，得出的结果也是浮点数；

扩展：

% ：可把比较大的数据范围，按一定的方式，缩小到指定的范围；

**例**：将随机数定在一个指定的数据范围内。

1. 比较运算符|关系运算符

**< > <= >= = !=**

1. 逻辑运算符

**&&（and）、||（or）、!(not)**

# 六、条件控制语句

## if语句

表达式：

if(条件满足){

//做什么事情; 🡪执行语句

}else{

//做其它事情; 🡪执行语句

}

**备注**： if语句最好都带上else，即使不满足if条件就不做任何处理，也在else里面注明不做任何处理；

## 多重if语句

表达式：

1. if(选择条件1){

//满足条件1则执行此处的执行语句；

}else if(选择条件2){

//！条件1&&条件2则执行此处的执行语句;

}else{

//！条件1&&！条件2则执行；

}

1. if(条件1){

//满足条件1执行此处语句；

}

If(！条件1&&条件2){

//！条件1且条件2则执行；

}

**备注**：建议不要用B的方式写代码，代码复杂，且计算运行效率低；

## 嵌套if语句

条件表达式代码块里再套一个条件表达式。

表达式：

if(判断条件1){

if(判断条件2){

}

}

## switch语句

表达式：

switch(key)

case 常量1：{

}break; 🡪break:中断该语句的判断

case 常量2：{

}break;

…

default{

}

Switch 语句需注意的问题：

1. Case的数据类型必须与Key表达式相同；
2. Case的值不能相同；
3. Case的执行代码，最后要break中断；
4. Default语句可放在case语句之前，也可以放在最后；
5. 没有对应的常量值，就跳转到默认的语句（default）;
6. Case与常量之间要有空格；
7. 变量key的类型范围：char、bit、short、Int、String(jdk1.7之后的版本能用)

# 七、循环语句

1. while循环

初始化语句、比较语句、变量叠代；

While(比较语句){

//循环体；需要重复执行的代码；

//添加退出机制（步长：叠加的步数）

//或者用break退出;

}

1. do…while循环

//先执行一次，再判断是否继续循环

// while: 先判断是否继续循环，满足条件才执行

dowhile最后的分号“；”不要漏掉；

do{

//执行语句；

中断机制；

}while(比较语句);

**循环语句，也可以嵌套；**

1. for循环

把数据初始化、比较语句、叠代语句，放在一起；

for(数据初始化 **;** 比较语句 **;** 叠代语句){

//循环的内容

}

**break :** 中断或者退出循环体;

**continue ：** 本次循环中，关键字后面的不执行，直接转到下次循环;

# 八、数组

1. **一维数组**

方便管理大量数据（变量）

数据类型 [] 数组名； //申明；

两种初始化

**基本初始化 :**

* New 数据类型 [数组长度];

数组中，没有任何数据值，只默认值 ；

**带数据的初始化:**

* New 数据类型[]{数据1，数据2};

此种情况，不用指定长度，使用后面数据个数作为长度；

给数组赋值:

**数组名[索引|下标] = 数据**

1. **二维数组**

**数据类型[][] 数组名=new 数据类型[数组长度][];**