基础语法

package a;

public class 基础语法 {

/\*

\* Java是Sun公司在1995年5月正式推出的面向对象的编程语言.在2010年被Oracle公司收购.

\*

\* Java体系:

\* JavaSE: 标准版,主要为Java核心语法

\* JavaEE: 企业版,主要为Java Web开发

\* JavaME: 微型版,主要为Java手机应用开发.

\*

\* Java环境:

\* 实质为JDK软件的安装和环境变量位置

\* 环境变量: PATH=JDK安装目录\jdk\bin JAVA\_HOME=JDK安装目录\jdk\

\* JDK安装: jdk部分安装[第一个安装路径],jre部分安装[第二安装路径] --->百度参考

\*

\* 开发工具: eclipse myeclipse netbeans 等等

\* 说明: eclipse安装完成后,必须提供Java环境,才能启动. myeclipse自带JDK环境,因此不需要自己配置

\*

\*

\* 开发步骤:

\* 1. 新建java文件,文件后缀为 .java

\* 2. 在java文件编写java类: java类是java文件的基本格式

\* public class 类名{

\*

\* }

\* 注意:

\* 1.类名必须为当前文件名一致.

\* 2.所有java代码都必须编写在类体中 [{}称为类体]

\* 3. 编写程序执行入口: main方法

\* public static void main(String[] 参数名){

\*

\* }

\* 注意:

\* 1.main方法作为程序的执行入口,方法格式基本不变.[只能含有1main]

\* 2.大小区分

\* 3.所有具体的代码/命令必须编写方法体.

\*

\* 4. 编写输出语句

\* System.out.print("输出内容");

\* System.out.println("输出内容");

\*

\* 5. 执行Java程序: 编译 解析

\* 进入dos窗口,进入java文件所在目录

\* 编写命令: javac java文件名.java

\* 编写命令: java class文件名

\*

\* 强调: java属于编译解释型语言.[先编译(javac.exe),再解析{java.exe)]

\* 一次编译,到处运行.[跨平台性好]

\*

\*

\*

\*/

/\* 常规语法:

\* 注释: 单行注释 多行注释 文档注释

\* java程序会自动忽略命令行中的多余空格和换行符.

\*

\*

\* 基本语法:

\* 数据类型 变量 运算符 数组 控制语句 循环语句 转移字符

\*

\* 数据类型: 基本数据类型[8种基本数据类型] 引用数据类型[数组 接口 对象]

\* 注意:

\* 基本数据类型 传递的数据的值,引用数据类型传递的是内存地址.

\*

\* 变量: 变量是Java程序中最基本的存储单元,可以存储1个数据.

\* 变量格式:

\* 变量类型 变量名=值 或 变量类型 变量名; 变量名=值;

\* 注意:

\* Java属于强类型的语言,变量中只能存放相应类型的数据.[JavaScript是弱类型的语言.]

\* 全局变量: 在所有方法体之外创建的变量,称为全局变量.

\* 局部变量: 在方法体中创建的变量,称为局部变量. [方法体内变量,参数]

\* 就近原则: 同JS

\*

\* 运算符: + - / \* % >= <= != ==

\* 注意: "="表示赋值,"=="表示判断数据是否相等

\*

\* 数组: 格式与JS相同,区别在数据类型

\* 数组类型[] 数组名=new 数组类型[元素个数]

String[] i=new String[5];

\* 数组类型[] 数组名=new 数组类型[]{元素1,元素2,..}

\* 数组类型[] 数组名={元素1,元素2,..}

\*

\* 控制语句: if语句 switch语句 [同JS]

\* if(条件){

\* 当条件为true时,执行此处代码

\* }

\*

\* if(条件){

\* 当条件为true时,执行此处代码

\* }else{

\* 当条件为false时,执行此处代码

\* }

\* 同理 if else if else if else.

\*

\*

\* switch(表达式){

\* case 值1: 代码1;break;

\* ......

\* [default: ...]

\* }

\*

\* 循环语句: for while do while [同JS]

\* for(循环初始值;循环条件;循环控制条件){

\* 循环体

\* }

\* while(条件){

\* 循环体

\* }

\* do while同理.

\*

\* 转移字符: continue break return [同JS]

\*

\*

\*/

//单行注释

/\*

\* 多行注释

\*/

/\*\*

\* 文档注释: javadoc.exe可以利用文档注释,生成关于程序的说明文档.

\*/

public static void main(String[] args) {

}

}

扫描器

package a;

import java.util.Scanner;

public class 扫描器 {

/\*

\* Java程序中经常需要人机交互,Java系统提供1个扫描器类,可以进行基本的交互操作.

\* java.util.Scanner类: 在java/util包中,含有Scanner类.在使用时,必须告诉系统要使用该类:

\* import java.util.Scanner;

\*/

public static void main(String[] args) {

Scanner sc=new Scanner(System.in);//System.in控制输入源为键盘输入（scan alt+?）

System.out.println("请输入您的姓名:");

String name=sc.next();//生成1个扫描字符串的扫描器,并准备扫描

System.out.println("姓名:"+name);

//nextInt() nextFload() .... nextBoolean()

}

}

补充：快捷键

main alt+? main方法

syso alt+? 输出方法

scan alt+? 扫描器