* **命令提示符（cmd）**

快捷键启动 win+R

切换盘符 盘符名称：

进入文件夹 cd 文件夹名称

进入多级文件夹 cd 文件夹1\文件夹2\

返回上一级 cd ..

直接回跟路径 cd \

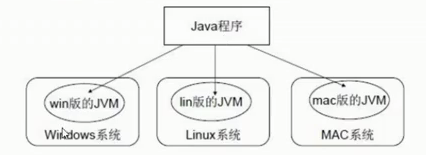
查看全部文件夹 dir

清屏 cls

退出 exit

* **Java跨平台性--JVM（Java虚拟机）**

JVM：Java虚拟机，语言件相互转换。Java的虚拟机（JVM）本身不具有跨平台性，每个操作系统下都有不同版本的虚拟机



* **JRE**

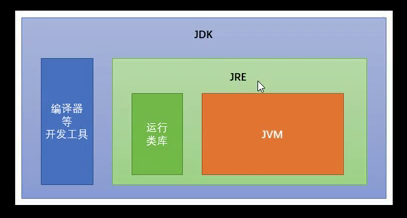
Java程序的运行环境，包含**JVM**和运行时所需的**核心类库**

只运行一个已有的Java程序，只安装**JRE**即可

* **JDk**

Java程序开发工具包，包含**JRE**和开发人员使用的**工具**

开发一个全新的Java程序，必须要安装**JDK**



* **注释**

单行注释： //

多行注释(区块注释)/\* \*/

* **Hello world**



Main方法：

Public static void main(String[] args){ ... }

* **关键字**：全部小写
* **标识符**

命名规则:

1. 包含英文字母（区分大小写）、0-9数字、$（美元符号）和\_
2. 不能以数字开头
3. 不能是关键字

命名规范：

1. 类名规范：首字母大写，后面每个单词首字母大写（大驼峰式）。
2. 变量名规范：首字母小写，后面每个单词首字母小写（小驼峰式）。
3. 方法名规范：同变量名。

* **常量输出**

1. system.out.println(‘’); 错！必须有且仅有一个字符，没有不行
2. System.out.println(‘AB’); 错！必须有且仅有一个字符，有两个不行
3. 空常量不能直接用来打印输出

* **数据类型**

分为基本数据类型和引用数据类型

引用数据类型：字符串、数组、类、接口、Lambda......

基本数据类型：除了引用数据类型之外的就是基本数据类型

* **基本数据类型**（四类八种）

整数型 byte short int long

浮点型 float double

字符型 char

布尔型 boolean

**注意事项**：

1. 字符串不是基本数据类型，而是引用类型
2. 浮点型可能只是一个近似值，并非精确的值
3. 数据范围与字节数不一定相关，例如float数据范围比long更广泛，但是float是4字节，long是8字节
4. 浮点数默认为double，若要用float，需要在后面加后缀F
5. 整数默认是int，若要用long，需要在后面加上后缀L

* **变量注意事项**

1. 如果创建多个变量，那么变量之间的名称不可以重复。
2. 对于float和long类型来说，字母后缀F和L不要丢掉。
3. 如果使用byte或者short类型的变量，那么右侧的数据值大小不能超过左侧数据类型的大小
4. 变量必须要在赋值后进行使用
5. 变量使用不能超过作用域的范围
6. 可以通过一个语句来创建多个变量，但是一般情况不推荐这么写。

* **数据类型转换**

**自动类型转换（隐式）**

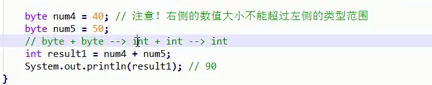
1. 特点：代码不需要进行特殊处理，自动完成
2. 规则：数据范围从小到大（tip：float的范围比long更大一些）

**强制类型转换（显式）**

1. 特点：代码需要进行特殊的格式处理，不能自动完成。
2. 格式：范围小的类型 范围小的变量名 = （范围小的类型）原本范围大的数据

**注意事项：**

1. 强制类型转换一般不推荐使用，因为有可能发生精度损失，数据溢出。
2. byte/short/char这三种类型都可以发生数学运算，例如加法“+”。
3. byte/short/char这三种类型在运算时，都会被首先提升为int类型，然后再计算



1. Boolean类型不能发生数据类型转换

* **算数运算符**

**注意事项**：

1. 一旦运算中有不同类型的数据，那么结果将会是数据类型范打的那种
2. 只有变量才能使用自增、自减运算符。常量不可发生改变，所以不能用

**“+”常见的三种用法：**

1. 对于数值来说，就是加法
2. 对于字符char来说，在计算之前，char会被提升为int，然后再计算。
3. 对于字符串String（首字母大写，并不是关键字）来说，加号代表字符串连接操作。任何数据类型和字符串进行连接操作时，结果都会变成字符串

* **赋值运算符（= += -= .....）**

**注意事项：**

1. 只有变量才能使用赋值运算符，常量不能进行赋值
2. 符合赋值运算符隐含了一个强制类型转换

* **比较运算符**

**注意事项：**

1. 比较运算符的结果一定是一个boolean值
2. 如果进行多次判断不能连着写，如1<x<2,需要用到 && 或 ||

* **逻辑运算符**

与（并且） &&

或（或者） ||

非（取反） ！

**注意事项**：与“&&”，或“||”，具有短路效果，如果根据左边已经可以判断得到最终结果，那么右边的代码将不再执行，从而节省一定的性能。

三元运算符

格式： 数据类型 变量名称= 条件判断 ？ 表达式A : 表达式B；

注意事项：

1. 必须同时保证表达式A和表达式B都符合左侧数据类型的要求。
2. 三元运算符的结果必须被使用。

* **Switch语句使用的注意事项**

1. 多个case后面的数值不可以重复
2. Switch后面小括号当中智能是下列数据类型：

基本数据类型：byte/short/char/int

引用数据类型：String字符串、enum枚举

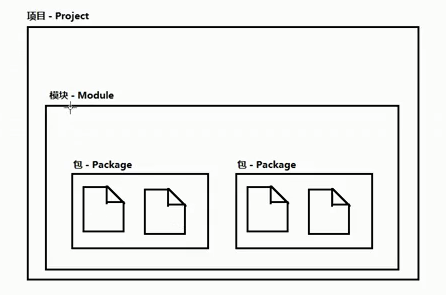
1. Switch语句格式可以很灵活：前后顺序可以颠倒，而且break语句还可以省略。“匹配哪一个case就从哪一个位置向下执行，直到遇到了break或者整体结束为止”

* **三种循环的区别（for while do-while）**

1. 如果条件判断从来没有满足过，那么for循环和while循环将会执行0次，但是do-while循环会执行至少一次
2. For循环的变量在小括号当中定义，只有循环内部才可以使用，while循环和do-while循环初始化语句本来就在外面，所以出来循环之后还可以继续使用。

* **Idea使用**

1. 项目模块



1. Package的名字要求为英文小写字母和英文句号（英文句号划分文件夹层次）
2. 快速写出主函数：psv/psvm

快速写出输出函数：sout

1. 调用控制台：alt+4
2. 补全提示由ctrl+空格，改成alt+/

* **Idea常用快捷键**

1. alt+enter 导入包，自动修正代码
2. ctrl+alt+L 格式化代码，调整排版
3. ctrl+Y 删除光标所在行
4. ctrl+D 复制光标所在行，插入在下面
5. Ctrl+/ 单行注释，多行注释，再按取消注释
6. Ctrl+shift+/ 选中代码注释，多行注释，再按取消
7. Alt+insert 自动生成代码，toString,get,set等方法
8. Alt+shift+上下剪头 移动当前代码行

* **方法重载**

与下列因素相关：

1. 参数个数不同
2. 参数类型不同
3. 参数的多类型顺序不同

与下列因素无关：

1. 与参数的名称无关
2. 与方法的返回值类型不同

* **Java中的内存划分**



* **一个数组的内存图**

