## 使用print() 打印：hello world

## 使用print() 打印6 + 9的和

## 使用print() 打印5与6乘积

## 使用print() 打印9除以3的结果

## 使用print() 打印3>5的结果，并思考这个结果代表什么

## 用翻译工具查询一下google、是什么意思？

谷歌，搜索引擎

## python与java,c语言的区别？

查询.C语言是编译语言，而Python和Java是解释型语言，  
编译语言在运行的时候通过预处理、编译器编译等转化成机器语言（一种低级语言），生成可执行文件，运行的效率高，速度快；而解释型语言则是在程序****每次****运行的时候一句一句把程序运行，所以效率相比较偏低运行慢。相比较而言，C语言更加适合去写追求运行效率的和底层运行有关的东西

3 .常用IDE：C语言是VS， Python是Pycharm， Java是IDEA和Eclipse

4 .Java 编译成字节码（.java文件经过javac的处理编程.class文件，即编写成字节码形式，再通过在不同的平台上通过各自的解释器解释执行），需由Java虚拟机（JVM）读取运行， 所以效率偏低，但是其可跨平台运行

5 .垃圾回收方面，C需要程序员自己回收，对内存进行良好的管理，而Java则不需要程序员自己完成这个工作，有自己的垃圾回收机制。

6 .在书写风格方面，我觉得最优雅、随意、简介的是Python，它甚至不需要你去在每一行语句后面加上分号，作为动态语言，Python非常的灵活，使用变量的时候不需要进行类型声明，里面的变量就像标签一样，想用在什么数据上，“贴上去”就行了；C语言朴实严谨，需要非常强的逻辑思考能力，作为现在广大高校计算机类专业的入门语言，非常适合学生对于编程思想和逻辑的锻炼，而Java则是一个纯面向对象的语言，以类为程序的主体，层次清晰，同时也是强类型语言（C语言和Python也是*强类型语言*），要求变量的使用要严格符合定义，所有变量都必须先定义后使用，同时的话，它对于自身每种数据类型的定义也做到了非常严格，比如：byte只能占用1字节的内存空间，short只能2字节，int只能4字节，而不是像C语言那样，每种数据类型占据多大的内存空间是看使用的编译器的，同时像Java的布尔型（boolean），也不能像C语言和Python那样，用数字0代表逻辑值“False”，1代表“True”，同时，像在C和C++中常见的指针类型在Java中为了避免空指针野指针等等产生混乱的情况是被取消了的，转而代之的是其独特的引用类型，如数组、类（class）、接口

再一个，是Python的标准库和第三方库非常强大，并且也很有趣，关键是下载也很方便（只要在命令行控制符中输入pip install （第三方库名）就行了），其中不乏一些强大的、现在被广泛使用的numpy，pandas，matplotlib等等这些用于数据分析作图的科学计算库、数据可视化模块，也不乏一些比较容易上手和有趣的如pygame（用于制作Python小游戏），bs4库（用于网络爬虫的制作），turtle（标准库，海龟作图），easygui（简单人机交互界面），pillow（用于图像处理），用于操作系统的os、sys、platform库等等。所以Python是易上手也很有趣。同时，Java也有很多的强大的库和API，如：HTML解析库、XML解析库、Http库、JSON解析库、序列化库、网络库、Excel读写库等等。

8 .说说它们的应用，  
C语言：C理论上说可以做任何开发, 只要有合适的硬件驱动和API，特点是效率高，基本上是编译语言里面效率最高的。目前而言, C语言主要用来开发底层模块(比如驱动,解码器,算法实现), 服务应用(比如web服务器)和嵌入式应用(比如微波炉里的程序)  
Java: Java常常跟"企业"联系在一起, 因为具备一些很好的语言特性, 以及丰富的框架, 在企业应用中最被青睐, 你总可以听到关于J2EE, JSP, Hibernate之类的东西的讨论. Java其本身就分为三大平台：Java SE（Java 2 Platform Standard Edition）、Java EE(Java 2 Platform Enterprise Edition)、Java ME(Java 2 Platform Micro Edition).Java在手机领域也有一席之地, 在普遍智能化之前, 很多手机就是以支持java应用作为卖点的, 而智能手机爆发之后, Java手机主场变成了android, 作为安卓的标准编程语言而存在。主要用途： 网页, 企业级开发, 普通应用软件、应用程序, 游戏后台、服务器开发、Web开发和安卓开发等。

## 查询一下什么是面向过程，什么是面向对象？

面向过程就是分析出解决问题所需要的步骤，然后用函数把这些步骤一步一步实现，使用的时候一个一个依次调用就可以了；面向对象是把构成问题事务分解成各个对象，建立对象的目的不是为了完成一个步骤，而是为了描叙某个事物在整个解决问题的步骤中的行为。

可以拿生活中的实例来理解面向过程与面向对象，例如五子棋，面向过程的设计思路就是首先分析问题的步骤：1、开始游戏，2、黑子先走，3、绘制画面，4、判断输赢，5、轮到白子，6、绘制画面，7、判断输赢，8、返回步骤2，9、输出最后结果。把上面每个步骤用不同的方法来实现。

如果是面向对象的设计思想来解决问题。面向对象的设计则是从另外的思路来解决问题。整个五子棋可以分为1、黑白双方，这两方的行为是一模一样的，2、棋盘系统，负责绘制画面，3、规则系统，负责判定诸如犯规、输赢等。第一类对象（玩家对象）负责接受用户输入，并告知第二类对象（棋盘对象）棋子布局的变化，棋盘对象接收到了棋子的变化就要负责在屏幕上面显示出这种变化，同时利用第三类对象（规则系统）来对棋局进行判定。

可以明显地看出，面向对象是以功能来划分问题，而不是步骤。同样是绘制棋局，这样的行为在面向过程的设计中分散在了多个步骤中，很可能出现不同的绘制版本，因为通常设计人员会考虑到实际情况进行各种各样的简化。而面向对象的设计中，绘图只可能在棋盘对象中出现，从而保证了绘图的统一。

## 编程实现展示xxx服装商城系统的衣服，并进行总金额的结算。