**使用minikube部署微服务案例系统：Online Boutique**

**Online Boutique**

在线时装店是一个微服务演示应用程序，它由11层微服务应用程序组成，是一个基于Web的电子商务应用程序。用户可以在其中浏览商品，将它们添加到购物车，然后购买。

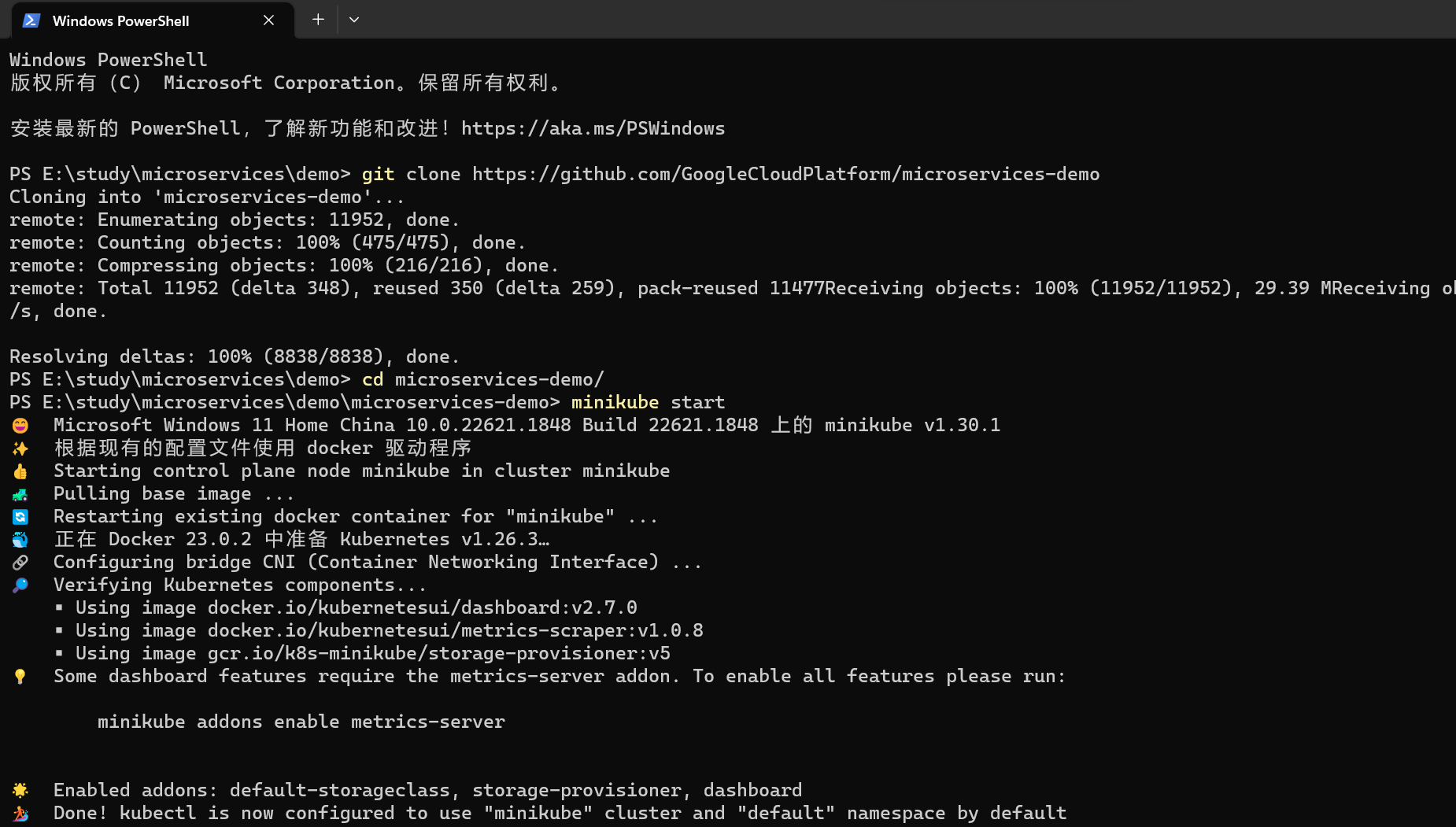
1. **将项目克隆到本地**

git clone https://github.com/GoogleCloudPlatform/microservices-demo

cd microservices-demo/

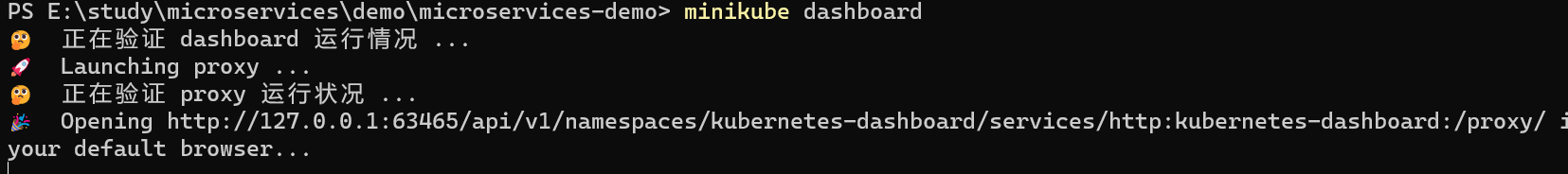
1. **打开minikube**

minikube start



1. **部署dashboard监控运行状况（新终端）**

minikube dashboard

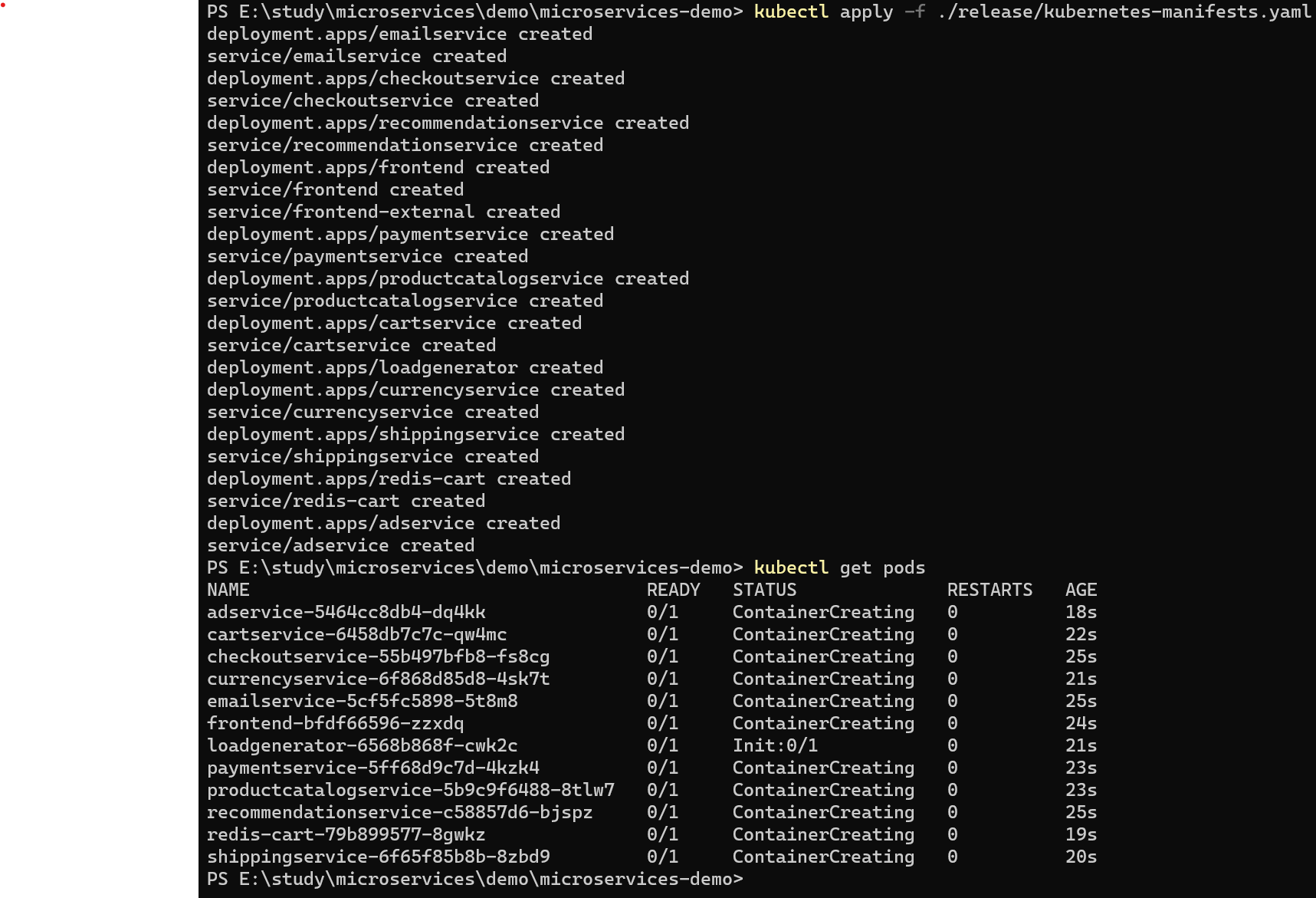


1. **部署案例系统**

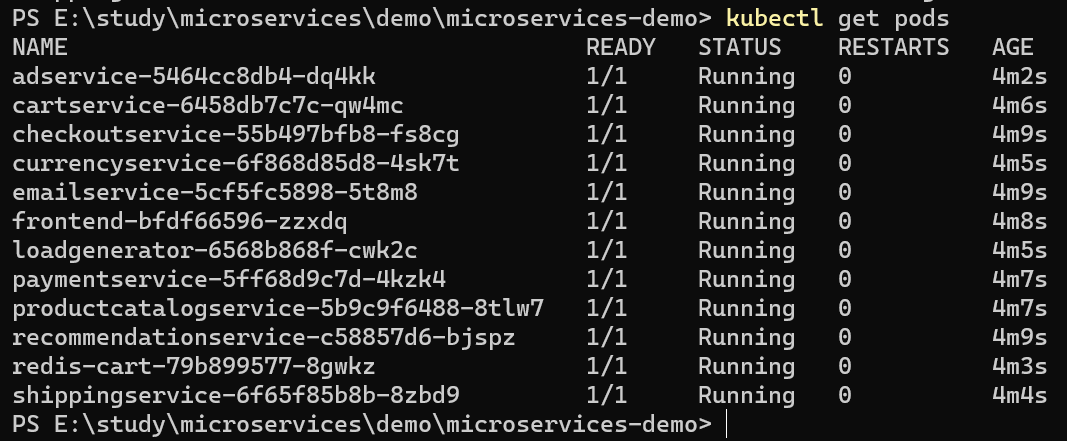
kubectl apply -f ./release/kubernetes-manifests.yaml

1. **等待pod准备就绪**

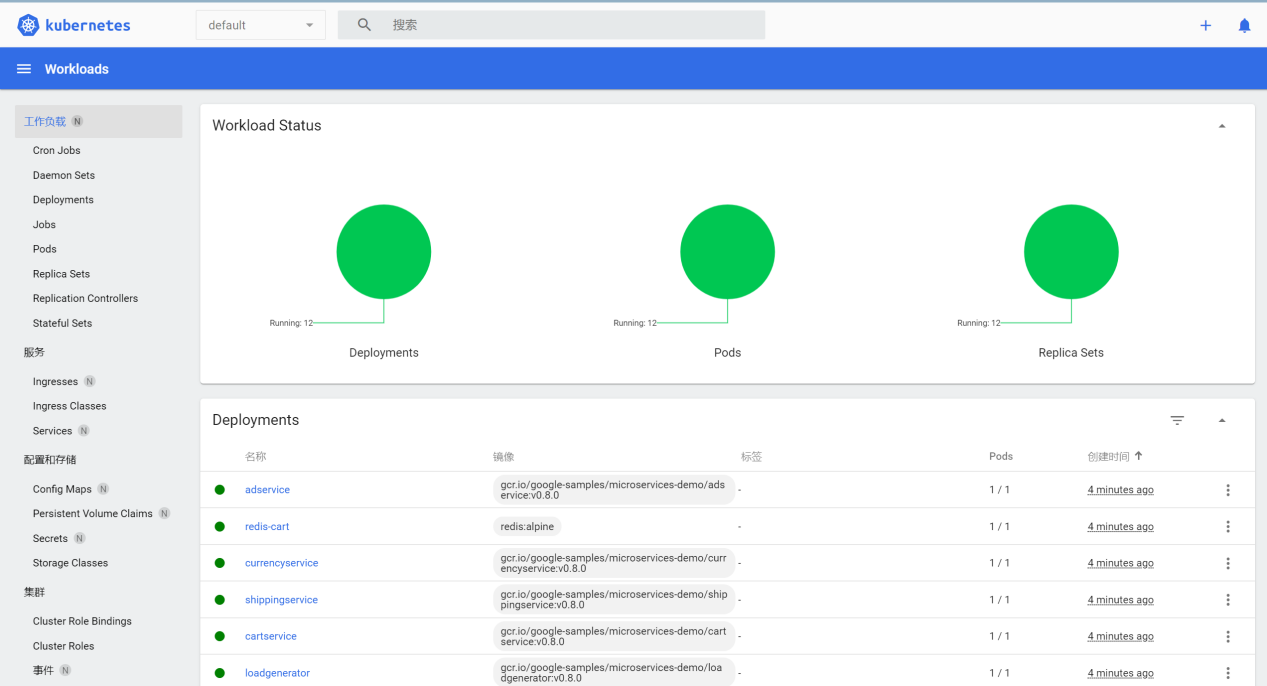
Kubectl get pods



等待几分钟后，发现pod处在running状态：

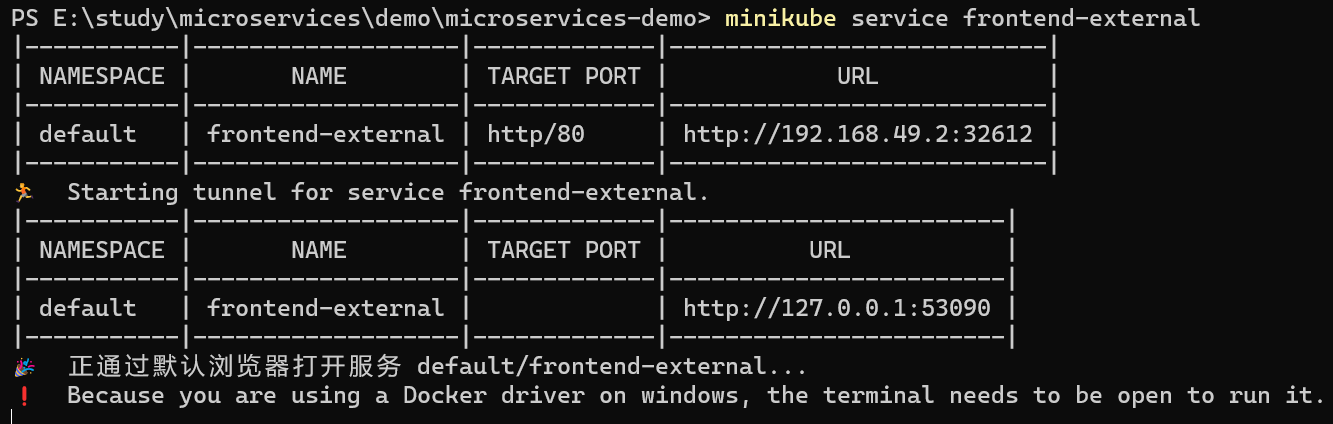


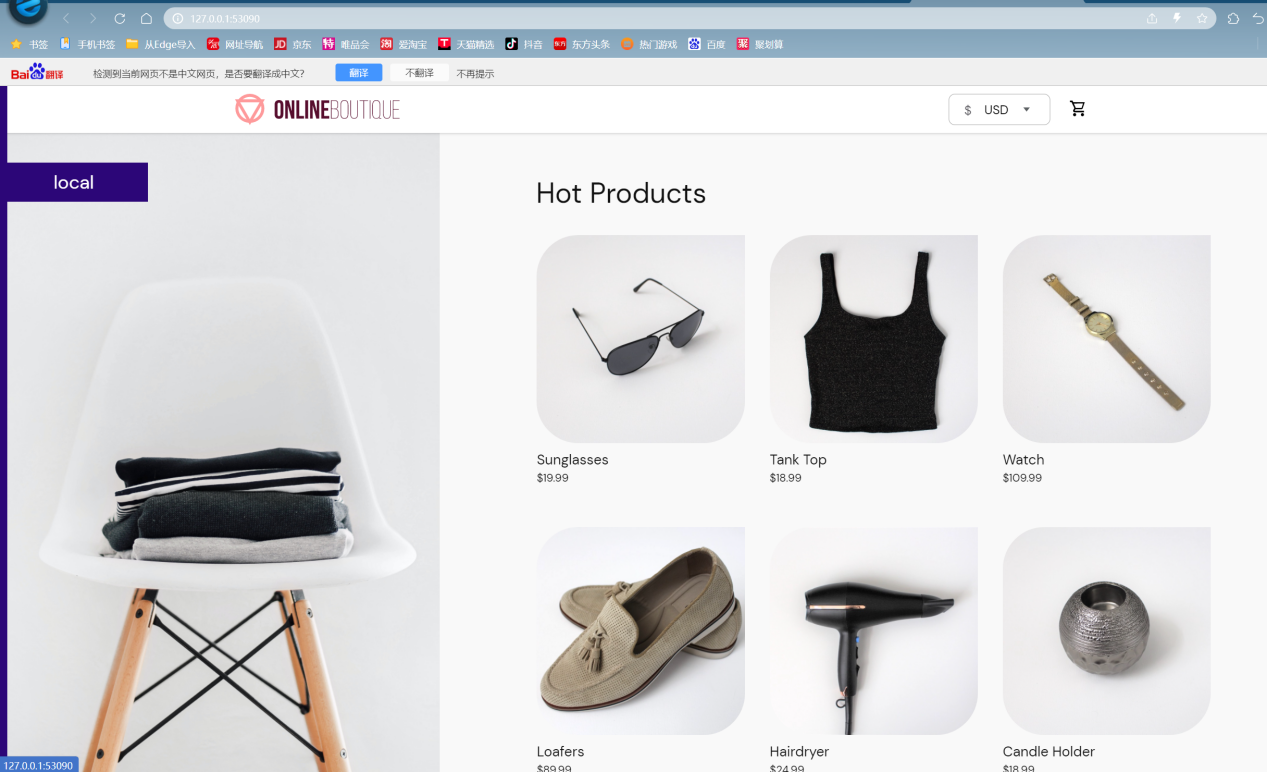
Dashboard可看到微服务的运行状态：



1. **在浏览器中访问web前端页面**

minikube service frontend-external





**复习java**

JDK：java开发者工具

JRE：java运行时环境

JVM：java虚拟机

**Java基础语法**

**注释**

单行/多行/文档

//单行注释

/\*多行注释 \*/

文档注释JavaDoc

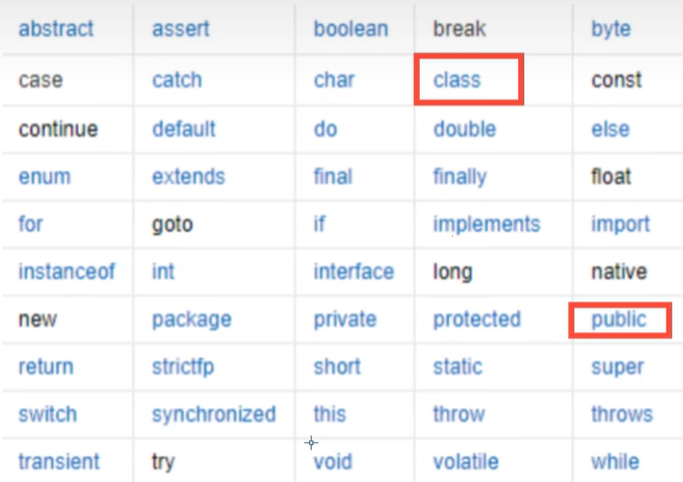
/\*\*

\*@Description HelloWorld

\*/

**标识符**

关键字



类名、变量名、方法名

标识符注意点：

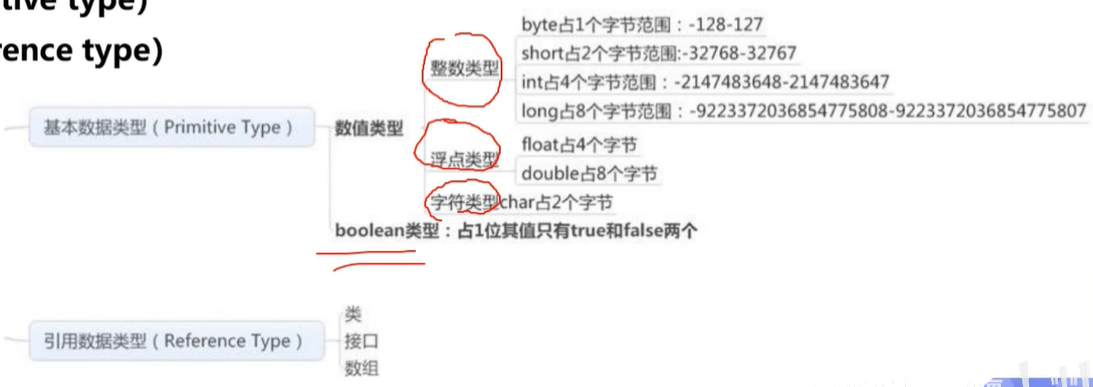
字母、$、下划线开头

大小写敏感

**数据类型**

强类型语言（变量使用要严格符合规定，先定义后使用）

基本/引用类型



Long类型加L，float类型加F

二进制0b 八进制0 十六进制0x

最好完全避免使用浮点数进行比较（有限、离散、舍入误差、接近但不等于）

转义字符\t\n

**类型转换**

强制（内存溢出、精度问题、“（类型）+变量名”、由高到低）

/自动（由低到高）

Jdk7新特性：数字之间可以用下划线分割，不输出（10\_0000）

**变量**

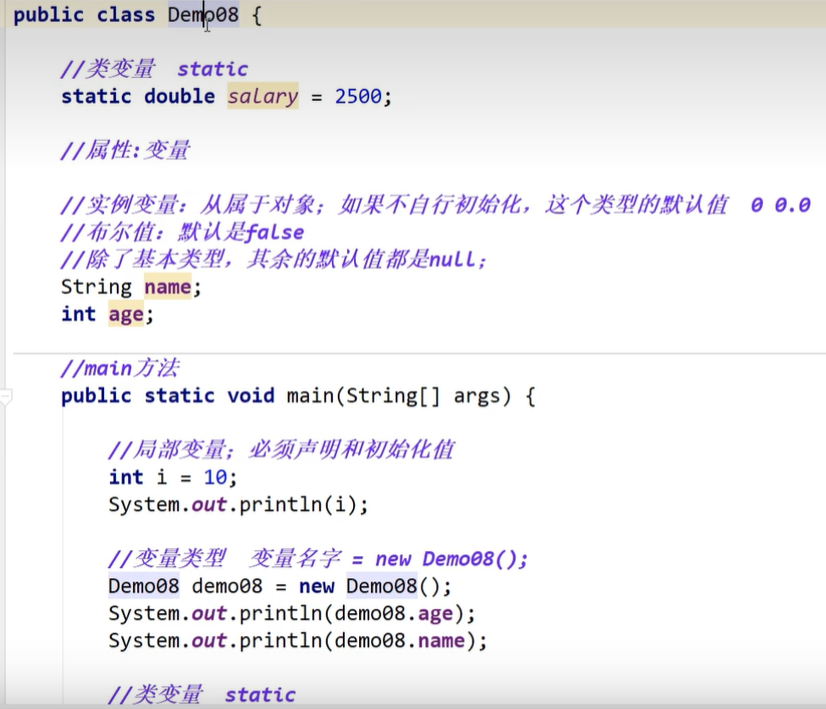
变量名、变量类型、作用域

建议一行声明一个变量

类变量/实例变量/局部变量

布尔值：默认为false





常量

final double PI=3.14;

命名规范：

类成员变量、局部变量、方法名：首字母小写和驼峰原则

常量：大写字母和下划线

类名：首字母大写和驼峰原则

**运算符**

\*\*ctrl+D：复制当前行到下一行

整数有long为long，无long为int

++a先加再执行

a++先执行再加

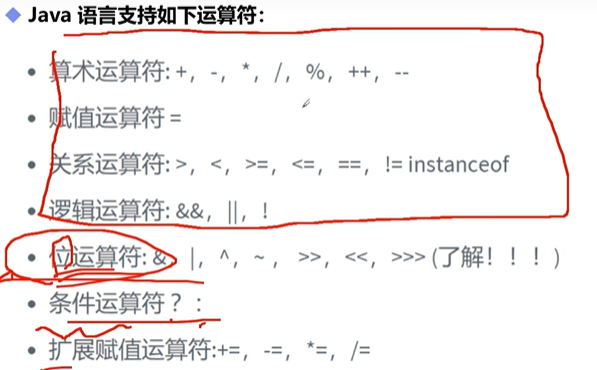
工具类

Math.pow(3,2)幂运算

短路运算：与或

2\*8=>2<<3左移

X ? y : z 如果x则结果为y，否则为z （代替if语句）



**学习spring**

**理念：**使现有的技术更加容易使用，本身是一个大杂烩，整合了现有的技术框架。

SSM：SpringMvc+Spring+Mybatis

官网：

<https://spring.io/projects/spring-framework>

官方下载地址：

https://repo.spring.io/release/org/springframework/spring

中文文档：

https://www.docs4dev.com/docs/zh/spring-framework/5.1.3.RELEASE/reference/

Github：

<https://github.com/spring-projects/spring-framework>

Maven仓库：<https://mvnrepository.com/tags/spring>

<!-- https://mvnrepository.com/artifact/org.springframework/spring-webmvc -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-webmvc</artifactId>

<version>6.0.10</version>

</dependency>

<!-- https://mvnrepository.com/artifact/org.springframework/spring-webmvc -->

<dependency>

<groupId>org.springframework</groupId>

<artifactId>spring-jdbc</artifactId>

<version>6.0.10</version>

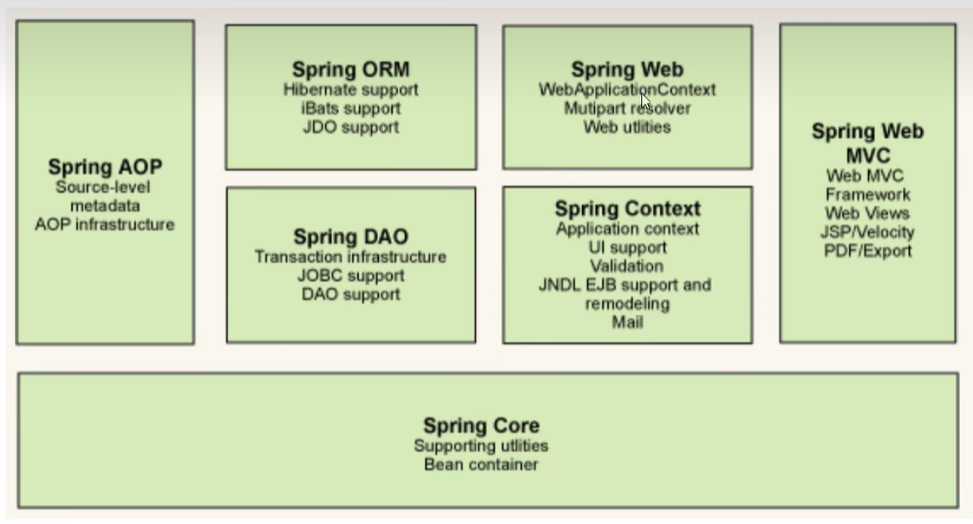
</dependency>

**优点：**

开源免费框架（容器）；轻量级、非入侵式；**控制反转（IOC）、面向切面编程（AOP）**；支持事务的处理、支持框架整合

**弊端：**配置繁琐

**组成：**



**学习路线：**



Spring、SpringMVC

**Spring Boot**

快速开发的脚手架

可以快速开发单个微服务

约定大于配置

**Spring Cloud**

基于Spring Boot实现

**IOC理论（控制反转）**

加入set注入，使程序的控制权从程序员手上转移到用户手上。程序员不用管理对象创建，降低系统耦合性，可以专注扩展业务。



控制反转是一种通过描述（XML或注解）并通过第三方去生产或获取特定对象的方式。在SPringle中实现控制反转的是IoC容器，其实现方法是依赖注入（DI）。

**springboot**

是什么

配置如何编写yaml

自动装配原理

集成web开发

分布式开发Dubbo+zookeeper

Swagger：接口文档

任务调度