# 面试题

* RabbitMQ是什么？
  + RabbitMQ是实现了高级消息队列协议（AMQP）的开源消息代理软件（亦称面向消息的中间件）。RabbitMQ服务器是用Erlang语言编写的，而群集和故障转移是构建在开放电信平台框架上的。所有主要的编程语言均有与代理接口通讯的客户端库。
* 使用RabbitMQ有什么好处？
  + 解耦，系统A在代码中直接调用系统B和系统C的代码，如果将来D系统接入，系统A还需要修改代码，过于麻烦！
  + 异步，将消息写入消息队列，非必要的业务逻辑以异步的方式运行，加快响应速度
  + 削峰，并发量大的时候，所有的请求直接怼到数据库，造成数据库连接异常
* RabbitMQ的特点?
  + 可靠性: RabbitMQ使用一些机制来保证可靠性， 如持久化、传输确认及发布确认等。
  + 灵活的路由 : 在消息进入队列之前，通过交换器来路由消息。对于典型的路由功能， RabbitMQ 己经提供了一些内置的交换器来实现。针对更复杂的路由功能，可以将多个 交换器绑定在一起， 也可以通过插件机制来实现自己的交换器。
  + 扩展性: 多个RabbitMQ节点可以组成一个集群，也可以根据实际业务情况动态地扩展 集群中节点。
  + 高可用性 : 队列可以在集群中的机器上设置镜像，使得在部分节点出现问题的情况下队 列仍然可用。
  + 多种协议: RabbitMQ除了原生支持AMQP协议，还支持STOMP， MQTT等多种消息 中间件协议。
  + 多语言客户端 :RabbitMQ 几乎支持所有常用语言，比如 Java、 Python、 Ruby、 PHP、 C#、 JavaScript 等。
  + 管理界面 : RabbitMQ 提供了一个易用的用户界面，使得用户可以监控和管理消息、集 群中的节点等。
  + 令插件机制: RabbitMQ 提供了许多插件 ， 以实现从多方面进行扩展，当然也可以编写自 己的插件
* AMQP是什么?
  + RabbitMQ就是 AMQP 协议的 Erlang 的实现(当然 RabbitMQ 还支持 STOMP2、 MQTT3 等协议 ) AMQP 的模型架构 和 RabbitMQ 的模型架构是一样的，生产者将消息发送给交换器，交换器和队列绑定 。
  + RabbitMQ 中的交换器、交换器类型、队列、绑定、路由键等都是遵循的 AMQP 协议中相 应的概念。目前 RabbitMQ 最新版本默认支持的是 AMQP 0-9-1。
* RabbitMQ有那些基本概念？
  + Broker： 简单来说就是消息队列服务器实体
  + Exchange： 消息交换机，它指定消息按什么规则，路由到哪个队列
  + Queue： 消息队列载体，每个消息都会被投入到一个或多个队列
  + Binding： 绑定，它的作用就是把exchange和queue按照路由规则绑定起来
  + Routing Key： 路由关键字，exchange根据这个关键字进行消息投递
  + VHost： vhost 可以理解为虚拟 broker ，即 mini-RabbitMQ server。其内部均含有独立的 queue、exchange 和 binding 等，但最最重要的是，其拥有独立的权限系统，可以做到 vhost 范围的用户控制。当然，从 RabbitMQ 的全局角度，vhost 可以作为不同权限隔离的手段（一个典型的例子就是不同的应用可以跑在不同的 vhost 中）。
  + Producer： 消息生产者，就是投递消息的程序
  + Consumer： 消息消费者，就是接受消息的程序
  + Channel： 消息通道，在客户端的每个连接里，可建立多个channel，每个channel代表一个会话任务