# 重要面试问题

分享一些我面试中的高频问题

## 类加载机制

## GET和POST区别

1.GET请求的数据会附在URL之后，以?分割URL和传输数据，参数之间以&相连， POST把提交的数据则放置在是HTTP包的包体中。

2.GET的长度受限于url的长度，而url的长度限制是特定的浏览器和服务器设置的，理论上GET的长度可以无限长。

3.POST是没有大小限制的，HTTP协议规范也没有进行大小限制，起限制作用的是服务器的处理程序的处理能力

https://www.zhihu.com/question/28586791

## 链表和数组的区别

## python的dict实现

https://blog.csdn.net/weixin\_43064185/article/details/107565845 https://www.cnblogs.com/muyiblog/p/7662262.html

## python和java的垃圾回收机制

https://blog.csdn.net/xiongchengluo1129/article/details/80462651 https://www.cnblogs.com/ArsenalfanInECNU/p/9077967.html

推荐：https://zhuanlan.zhihu.com/p/83251959

1. mysql的联合索引的结构

MySQL使用经验

1. rabbitmq高性能的原因

erlang，爱立信开发，一门专门为交换机软件开发诞生的编程语言

erlang特点 - 通用的面向并发的编程语言，适用于分布式系统 - 基于虚拟机解释运行，跨平台部署 - 进程间上下文切换效率远高于C语言 - 有着和原生Socket一样的延迟

（ATTENTION；这一块要重点看看）

总结 - RabbitMQ底层使用Erlang实现，天生具有高性能基因 - RabbitMQ在互联网和金融领域都有广泛的应用

总结2 - erlang进程间上下文切换效率远高于C语言和Java，进一步提高了RabbitMQ的并发性能 - erlang的网络性能有着和原生Socket一样的延迟，使得RabbitMQ的网络IO性能极高

1. rabbitmq的架构

Exchange总结 - AMQP协议直接决定了RabbitMQ的内部结构和外部行为 - 对于发送者来说，将消息发给特定的Exchange - 消息经过Exchange路由后，到达具体队列 - 消费者将消息从监听的队列中取走 - Exchange主要有3种类型：Direct / Topic / Fanout - Direct（直接路由）：Routing Key = Binding Key，容易配置和使用 - Fanout（广播路由）：群发绑定的所有队列，适用于消息广播 - Topic（话题路由）：功能较为复杂，但能降级为Direct，建议优先使用，为以后拓展留余地

1. celery的架构
2. 非递归写中序遍历

堆栈区别，垃圾回收

线程，进程，协程