**阻塞和非阻塞指的是执行一个操作是等操作结束再返回，还是马上返回。**

**比如餐馆的服务员为用户点菜，当有用户点完菜后，服务员将菜单给后台厨师，此时有两种方式：**

* **第一种：就在出菜窗口等待，直到厨师炒完菜后将菜送到窗口，然后服务员再将菜送到用户手中；**
* **第二种：等一会再到窗口来问厨师，某个菜好了没？如果没有先处理其他事情，等会再去问一次；**

**第一种就是阻塞方式，第二种则是非阻塞的。**

**同步和异步又是另外一个概念，它是事件本身的一个属性。还拿前面点菜为例，服务员直接跟厨师打交道，菜出来没出来，服务员直接指导，但只有当厨师将菜送到服务员手上，这个过程才算正常完成，这就是同步的事件。同样是点菜，有些餐馆有专门的传菜人员，当厨师炒好菜后，传菜员将菜送到传菜窗口，并通知服务员，这就变成异步的了。其实异步还可以分为两种：带通知的和不带通知的。前面说的那种属于带通知的。有些传菜员干活可能主动性不是很够，不会主动通知你，你就需要时不时的去关注一下状态。这种就是不带通知的异步。**

**对于同步的事件，你只能以阻塞的方式去做。而对于异步的事件，阻塞和非阻塞都是可以的。非阻塞又有两种方式：主动查询和被动接收消息。被动不意味着一定不好，在这里它恰恰是效率更高的，因为在主动查询里绝大部分的查询是在做无用功。对于带通知的异步事件，两者皆可。而对于不带通知的，则只能用主动查询。**

　　但是对于非阻塞和异步的概念有点混淆，非阻塞只是意味着方法调用不阻塞，就是说作为服务员的你不用一直在窗口等，非阻塞的逻辑是"等可以读（写）了告诉你"，但是完成读（写）工作的还是调用者（线程）服务员的你等菜到窗口了还是要你亲自去拿。而异步意味这你可以不用亲自去做读（写）这件事，你的工作让别人（别的线程）来做，你只需要发起调用，别人把工作做完以后，或许再通知你，它的逻辑是“我做完了 告诉/不告诉 你”，他和非阻塞的区别在于一个是"已经做完"另一个是"可以去做"。