1. **read、readline、readlines的区别**
2. read(size)方法从文件当前位置起读取size个字节，若无参数size，则一次性读取整个文件内容。它返回为字符串对象。
3. readline() 每次读取一行内容。内存不够时使用，一般不太用。所以，读取时占用内存小，比较适合大文件，该方法返回一个字符串对象。
4. **readlines() 方法读取整个文件所有行，保存在一个列表(list)变量中，方便遍历。每行作为一个元素，但读取大文件会比较占内存。**
5. **with open (‘文件名’) as f,for i in f: print(i) 使用for循环遍历文档内容,但有一个弱点，如果文档有异常数据不好处理，且不适用于大文档。**
6. **列出5个python标准库**

**os：提供了不少与操作系统相关联的函数**

**sys:   通常用于命令行参数**

**re:   正则匹配**

**math: 数学运算**

**datetime: 处理日期时间**

1. **迭代器和生成器的区别**

**迭代器定义：迭代器是一种支持next()操作的对象。它包含了一组元素，当执行next()操作时，返回其中一个元素。当所有元素都被返回后，再执行next()报异常—StopIteration  
生成器一定是可迭代的，也一定是迭代器对象。**

**生成器定义：生成器本质上就是一个函数，它记住了上一次返回时在函数体中的位置。  
它们写起来就像是正规的函数，只是在返回数据的时候需要使用yield语句。每次next()被调用时，生成器会返回它脱离的位置（它记忆语句最后一次执行的位置和所有的数据值）**

**区别：**

**①生成器是生成元素的，迭代器是访问集合元素的一种方式  
②迭代输出生成器的内容  
③迭代器是一种支持next()操作的对象  
④迭代器（iterator）：其中iterator对象表示的是一个数据流，可以把它看做一个有序序列，但我们不能提前知道序列的长度，只有通过nex()函数实现需要计算的下一个数据。可以看做生成器的一个子集。**

**4、数据库的三大范式是什么？**

**1、第一范式(1NF) ：列不可拆分**

**2、第二范式(2NF) ：唯一标识**

**3、第三范式(3NF) ：引用主键**

**后1个范式,都是在前1个范式的基础上建立的。**

**5、TCP粘包问题及解决？**

**TCP粘包是指发送方发送的若干包数据到接收方接收时粘成一包，从接收缓冲区看，后一包数据的头紧接着前一包数据的尾。**

**出现粘包现象的原因是多方面的，它既可能由发送方造成，也可能由接收方造成。**

**粘包原因：**

**TCP采用了流式传输，没有消息边界。对于数据频繁传输的程序来说，从接收端来看，收到的TCP传过来的数据可能就是一条两条三条数据连在一起，或者断头断尾不完整的。粘包原因是多方面的，可能是发送端的问题也可能是接收端的问题，可能是程序原因也可能是网络延迟等等。**

**解决办法：**

**1、发送固定长度的消息**

1. **使用特殊标记来区分消息间隔，包尾加上\r\n标记，设置消息边界。**

**3、通过在两次send之间调用sleep来休眠一小段时间来解决。**