1.python with 如何使用？好处是什么？为什么可以起到关闭文件的作用？

使用场景：对于 python 在写入文件或者是读取文件的结束，需要手动关闭文件，极端的情况会出现 too may open files 的错误，因为系统允许打开的文件数量是有限的。

代码演变:

初级：下面是不使用 with 的代码，会出现的问题是当使用 readline 出现异常的时候，下面的代码就不会被执行，因而文件就不会被关闭。

```
f = open("test.txt",'r')
f.readline()
f.close()
```

进阶：下面的代码使用了 try 进行异常捕捉，如果出现异常则会运行 except 代码，最后执行 finally 代码，该文件一定会被关闭，

```
f = open("test.txt",'r')
try:
    f.readline()
except iOError:
    print('error')
finally:
    f.close()
```

高级：下面的代码系统会自动调用 close 方法，所以和进阶的作用是一样的，而 with 实现的原理就是实现了上下文管理器。

```
with open("test.txt",'r') as f:
    f.readline()
```

上下文管理器：

任何实现了 __enter__()和__exit__()方法的对象都称为上下文管理器。因此可以使用 with 关键字，显然 file 类实现了上下文管理器。

手动写出 file 关闭文件功能：即使在 readline()函数出现异常，__exit__()也会被调用。

```
class file(object):
    def __init__(self, filename, mode):
        self.filename= filename
```

```
        self.mode = mode

    def __enter__(self):
        print('entering')
        self.f = open(self.filename, self.mode)
        return self.f

    def __exit__(self, exc_type, exc_val, exc_tb):
        print('will exit')
        self.f.close()

    if __name__ == '__main__':
    with file('test.txt','r') as f:
        1/0
        print(f.readline())
```

2. python __name__的作用？

概念：所有的模块都含有__name__属性，而__name__属性的值取决与如何使用这个模块，如一个模块为 learning_python.py

如果导入该模块：import learning_python，则__name__属性的值为 leanring_python

如果运行该模块：python2 leanring_python.py,则__name__属性的值为 main

作用：所以可以添加如下代码,如果该模块被导入，则__name__为模块名字，则该 if 不会被执行，如果直接运行该模块，则__name__为__main__则该 if 会被运行。

```
if __name__ == "__main__":
    do_something()
```

3. python is 和==的区别？

前言：python 一切都是对象，而对象就会有三要素，id，value 和 type

概念：

is 比较的是对象的 id 是否相同，也就是是否是同一个对象，是否指向同一个内存地址。

==比较的是对象的 value 值，

下面的结论只针对 python2.7

结论：

当变量为数字，字符串，元组，列表，字典的时候，==和 is 不能互换使用，比较数值的时候尽量使用==，当比较两个对象 id 是否相同可以使用 is

注：如果是数值，范围在[-5,256]则 is 和==的结果是一样的。