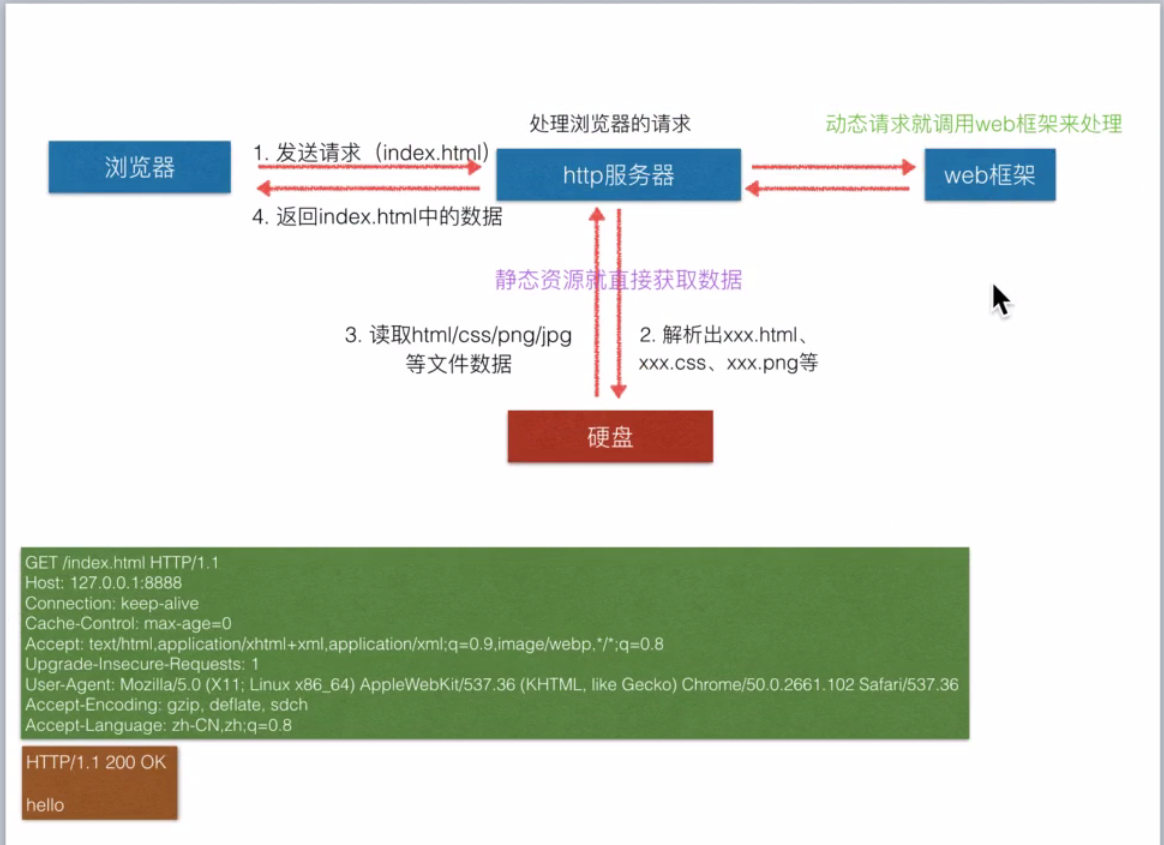
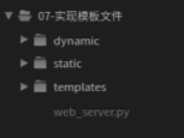
File\_name.endwith(‘.py’)

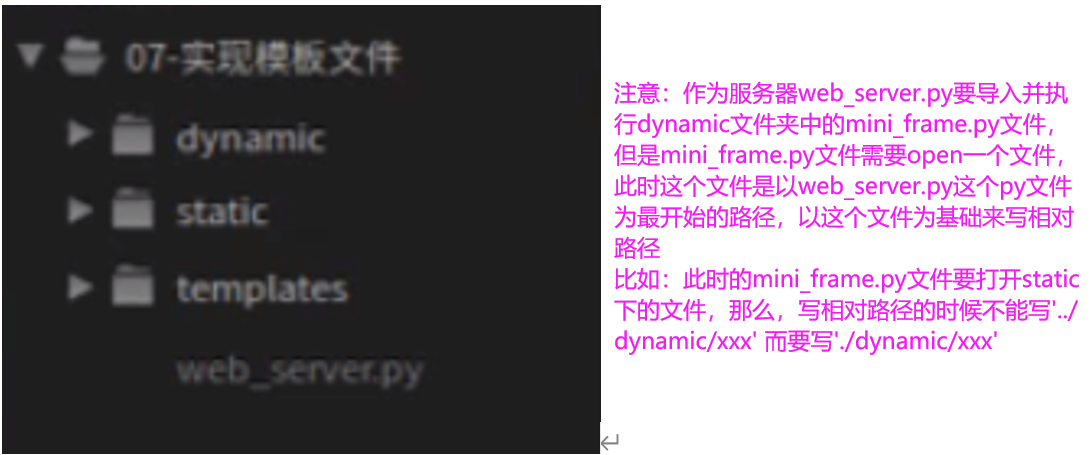
File\_name.startwith(‘.py’)

以上两个是分别判断file\_name是否以 ’.py’ 结尾/开头



让http服务器和我们写的功能代码（注册、登陆等等）分开，进行解耦





注意：再写代码的时候，框架和服务器要分离开来，可以新建一个文件夹，把它当作一个包，里面可以有框架所需要的所有模块

Sys模块中有个argv函数，这个函数可以返回给程序传递的参数

比如： python3 test.py hahaha 7890

那么返回的之是一个列表[‘test.py’,’hahaha’,’7890’]

因此，返回的列表里的内容是除了python3以外的所有内容，包括py文件名，而且全部以字符串的形式返回

Test.py文件的demo

Import sys

Print(sys.argv)

注意：以file\_name = mini\_frame为变量，我们要导入mini\_frame

如果通过import file\_name来导入模块的话，那么import导入的是file\_name这个名的py文件，那么文件夹中是没有的

但是有一个模块\_\_import\_\_()，括号里面输入一个变量名，那么此时，\_\_import\_\_(file\_name)就可以导入mini\_frame模块，\_\_import\_\_的返回值是一个对象，而这个对象就指向这对应的模块。还有一个getattr()函数，这个函数可以找到对应模块的对应函数，比如xxxxx = getattr(file\_name,xxxxxxxx),那么这个返回值就指向了对应的函数，file\_name为模块名，xxxxxxx为模块里的函数名

简而言之，import 后面不能导入变量名对应的文件，\_\_import\_\_可以

Sys.path.appenmd(……….)

以上这行代码，可以让代码添加某个路径，再搜索模块的时候，系统会自己再这个路径下查找

比如：写了个xxx.py文件，而且并没有再当前路径下，再./package路径下那么，再导入的时候，系统只会在默认的几个路径下查找，那么此时肯定就导入不了，如果用以上代码把该路径添加上去，那么就可以导入该模块了