递归

代码执行之后，回来再执行一段代码

把文件夹中的文件删除之后，再回来把这个文件夹删除就可以了

# 序列化

把对象写入到文件中

序列化

反序列化 把文件中的内容生成一个对象

js json.stringify

json.parse()

输入输出流

对象输入输出流

对象生成几种方式？

new

clone 克隆

反序列化

反射机制

js中 对象拷贝几种方式  
json.parse

… 运算符

存储在文件中

永久的意思

临时的

抖音 很多内容 很出名

# 打印流

printStream 字节

printwriter 字符

把所有的类型转化为字符

# 随机流

读写文件通过i/o是比较麻烦，如果里面数据要么读取全部的。要么写入全部的数据

可以使用i/o操作

对文件经常修改，增加，删除等操作

可以做到，麻烦

用户注册，登录，课程信息

课程修改，用户修改

申请课程

等等

讲完 类型，static 面向对象 集合list set map

做一个项目中的一个功能模块

迭代的粒度更小了。

内存版的

io流的时候，又可以做一个功能模块了

数据放在文件中了

文件版的

迭代 缺点，连贯性比较强，只要哪个环节学的不好，那么后面做不下去了。

粒度放大

还是迭代式 一个完整的功能模块一个迭代。

i/o

很重要，

基本昨天讲的都要理解

保存一个用户

1,zhangsan,man,22

2,lisi,woman,25

1万

要么获取所有的数据

遍历找到1号年龄22 -》23

把所有的数据写入到文件中

RandomAccessFile

随机访问

迭代一个功能

都是通过数据库 操作

mysql

其实也是文件操作

只是通过一种语言方便操作

第四代语言

第一代 机器语言

第二代 汇编

第三代 高级语言

第四代 sql 语言

id name sex age

insert into 表 values(1,zhangsan,man,22)

update 表 set age=23 where id=1

全部读和全部写

单一的操作 可以使用i/o

文件流的性能比数据库要高

数据库是对文件操作的一个包装

尤其在网络通讯上面，文件流经常使用

读全部，写全部

# 集合框架

数组其实也是集合 相同类型的东西放在一起的

集合框架 不一定，可以放各种类型

是数组的一个包装版

可以重复，也有顺序的，但是不能指定位置加入或者获取

子接口可以 list接口

我跟你们讲知识点，充其量 就是让你们快速入门，少走弯路

仅此而已

真正学到东西，自己去写代码，去理解，以后就不会忘记了。

jdk5之后

jdk版本低

jdk7

关键还有一个编译版本

性能