**项目要求：**

完成Django项目美多商城的总结，以小组为单位模拟真实项目开发，给小组每个成员分配任务模块，每个人将自己负责的项目模块进行总结，组长进行汇总，假期结束后对小组进行检查。

找三个非电商项目，最多一个电商衍生项目分析流程。

要求：整理项目业务流程、分析所找项目的数据表结构

参考网址：http://top.chinaz.com/



练习：

1:强类型语言和弱类型语言的区别,python是什么类型的语言(强类型,弱类型,解释型,编译型)

区别:强类型的的语言,变量的类型有严格的规范, 不存在类似 js 中的隐式转化这种. int和str等之间有明显的区别.

python是强类型,动态的,解释型的语言

2:is 和 == 的区别

is 是用来判断 python中两个对象是否共用同一块内存地址;

== 是用来判断两个对象的值是否相等

3:进程、线程、协程的区别

进程: 是操作系统分配资源的最小单位;

线程: 是系统调度的最小单位

协程: 是更小的线程 ????

通俗来说: 进程就像工厂的一个厂房; 线程就像厂房中的一条条流水线; 协程就像流水线上一个个的工人.

4:GIL锁是什么

GIL是全局解释器锁,是python语言中在使用CPython解释器出现的一个资源竞争现象,使得同时运行的线程只有一个.它的局限性在于在 计算密集型的操作时, python将不能发挥多线程和协程的优势; 但是在 I/O密集型的操作时,python还是能发挥多线程和协程的优势. 同时也有很好的解决办法:1.在 计算密集型 操作的时候, 可以针对这部分代码使用 非Cpython解释器; 2.这部分代码用其它语言来改写,比如C语言; 3.使用多进程来代替多线程或者协程.

5:MySQL常见数据库引擎及比较

两个: InnerDB 和 ISAMI

一般默认是 InnerDB , 可以设置 索引;

6:char和varchar的区别

都是数据库中对字符类型字段的一个表示方法; 但是varchar的字符是可变长度的字符,而char是不可变长度的字符.相比之下,前者更加节约资源消耗,相同情况下可以提高检索效率.

7:深拷贝和浅拷贝的区别

深拷贝和浅拷贝都是是对数据的一种复制操作.正如其名, 深拷贝是对内部的内容的拷贝,内存地址将改变, 而浅拷贝是拷贝的引用.

当然这个还要 对其数据的类型按照可变类型和不可变类型 需要更加深入的分析

我一般使用的分析方法是: 在遇到相关问题的时候, 可以通过 id 这个内置方法在ipython交互环境中来比较和判断,没有刻意去记忆这些

8:什么是lambda是函数,在工作中有什么作用

匿名函数, 可以简化代码,对于那些比较简单的函数可以通过lambda关键字定义并使用.

可以举出一些高端的例子:

9:如何分析SQL语句的执行计划？怎样查看正在执行的SQL？

查看日志

tail 命令(Linux系统中…)

10:git代码切换到上一次提交的状态

git reverst

11:什么是SQL注入，如何避免SQL注入

SQL注入: 由于sql语句不规范撰写和缺乏SQL语句关键词过滤,导致一些懂SQL语句的用户恶意侵入数据库的情况.

避免:

1. 制定SQL语句的撰写规范,比如对于特殊的 字符 需要加上 小引号(Tab上面的那个键)
2. 设计数据库表的时候,尽量将 字段名设置的比较难以猜出
3. 使用函数对简单的 sql 语句进行一定的封装 或者 使用优秀的第三方库来操作数据库
4. 对用户的操作进行 关键词 过滤

12:备份远程服务器(112.113.114.115)的数据库(端口为默认端口,帐号root,密码 mysql)到本地/home/python/Desktop下的命令是什么

source mysql -uroot -pmysql -h112.113.114.115 > myql -uroot -pmysql

13:如何将已经排序好的列表变为无序列表

import random

li = [I for I in range(10)]

random.shuffle(li)

li

14:python2和python3的区别,举例几个

1. 字符串编码: py2 中 字符串是 bytes 类型; py3 中 字符串是 str 类型
2. print函数语法的区别
3. 新式类和旧式类: MRO 的顺序()?? 还有定义: py3默认继承形式类,而且没有旧式类;py2中只有将继承自object的才是新式类,其它都是就是类

15:在项目中导包的顺序

通过导入sys, 在通过 sys.path 来查看

16:POST和GET的区别

1. get 没有请求体 ; post 有请求体
2. get 传输数据只能通过 url 其长度按照 HTTP协议约定不能超过 2048个字符,大小不能超过 1kb,但是不同浏览器可能有自己更严格的约束; post 支持上传图片不限制长度;
3. get 传输数据未加密; post传输数据加密,相对更加安全.

17:COOKIE和session的区别

COOKIE 保存在浏览器客户端; session 保存在服务器端;

session是基于cookie ;

18:L = [i for i in range(100)] 获取L中元素的个数的方法,向L中添加元素的方法,计算L个出现的某个参数出现的次数

len(L)

L.append() 追加; L.extend() 扩展添加,可以同时添加多个元素或一个序列元素会被拆开成多个元素; L.insert(位置, 元素) 可以在指定的索引位置添加元素.

19:判断字典中是否存在这个key的方法

key in dict

或者 key in dict.keys()

20:装饰器的作用以及项目中的应用

python进阶必备神技, 可以在不改变原来代码的前提下增加新的功能.

1. 做权限校验或者权限控制;
2. 做系统日志记录;

21:python获取系统当前时间

import time

time.time() 时间戳; time.ctime()

22:获取当前目录下所有文件夹

import os

os.chdir()

23:写一段代码实现输入和输出的字母大小写翻转;例如输入”AAbbccddEEFF”输出”aaBBCCDDeeff”

input\_str = input(‘请输入:’)

temp\_str = ‘’

for i in input\_str:

if i.lower() == i:

temp\_str += i.upper()

else:

temp\_str += i.lower()

print(temp\_str)

24:写一段代码统计用户输入的字符串中大写字母个数,小写字母个数,数字个数

三个思路:1. 原始的 chr 来做判断; 2.正则表达式

这里使用正则表达式演示:

import re

temp\_str = 'aongONWWNNG123'

print(len(re.findall(r'[a-z]', temp\_str)))

print(len(re.findall(r'[A-Z]', temp\_str)))

print(len(re.findall(r'[0-9]', temp\_str)))

25:写一段代码输出1000以内为7的倍数或者包含7的数字例如’14,17,21’

for i in range(1000):

if i % 7 == 0:

print(i)

elif '7' in str(i):

print(i)

26:生成器是什么,迭代器是什么,他们之间有什么区别

生成器是一种特殊的迭代器;

python中通过yield 这个关键字 来定义生成器; 当函数中包含 yield 的时候它将不在是一个普通函数而是一个生成器模板; 它会每次将yield 后面的结果返回,然后再继续从此处开始下一轮的 取值.

他们都是 python 设计用来 节约内存资源,而逐次从序列中取值的工具.迭代器是有规定的类,而且类中必须包含 \_\_iter\_\_(self)方法和\_\_next\_\_(self)方法,而且\_\_next\_\_(self)方法必须返回一个迭代器.

27:flask中请求上下文和应用上下文是什么

请求上下文: request session

应用上下文: current\_app 和 g变量

28:MVT的理解

一种重要的设计模式(思想):

M – model 对应数据库操作;

V – view 视图函数, 用来处理业务逻辑

T – template 模板 ,用来生成向前端返回的数据和格式

29:celery分为哪几块分别执行什么任务

分为: client(用来发送任务); broker(任务队列,用来存储任务); worker(任务执行者,监控任务队列的任务并进行异步执行)

30:你在项目中是否使用过select \* from table.

使用过, 但是一般不推荐直接使用 \* ; 一般查询会控制只查询有必要的字段和查询的条数.

31:mysql如何给账户设置权限

32:Git上传代码发生了冲突怎么处理

谁冲突谁处理,一般回退到前一个版本,或者pull下前一个版本,然后再修改后push.

33:L = [I for i in range(101)], 取出L中前10个元素,取出倒数10个元素,取出第10到40个元素的倒序并且为双数的.

L[0:10]

L[-1:-11:-1]

l = L[9: 40] # 牢记’左闭右开’; 对列表切片有涉及到拷贝

l.reverse() # 对于可变类型的数据来说, 反转操作修改的原来的数据内部.’链式调用’是针对有返回值的对象继续进行操作’

[i for i in l if i%2 == 0]

34:写一段代码实现数学中的阶乘,例如输入5,输出为5\*4\*3\*2\*1

num = 5

def factorial(n):

if n == 1:

fac = 1

else:

fac = n \* factorial(n-1)

return fac

35:请写一个函数:函数的输入是一个string类型的英文句子函数的输出也是一个string类型的应为句子：是把输入的句子里面的单词按逆序输出，但是每个单词按内部的字符按顺序输出示例：

输入： “I like you, but just I like you.”

输出： “.you like I just but, you like I”

注意事项：

标点符号当作一个单词处理

l = s.split(‘ ‘) # 用空格来分割,形成列表

l.reverse() # 列表逆序,注意因为该函数的返回值是空值,所以如果你链式调用那么接收到的数据就是None. 所以这一步和上面的那一步要分隔开,这个和前端不一样.一定要注意

‘ ‘.join(l) # 用空格拼接

36: 给定一个包含 n 个整数的数组nums，判断nums中是否存在三个元素 a，b，c ，使得 a + b + c = 0 ？找出所有满足条件且不重复的三元组。

Nums = [-1,0,1,2,-1,4]

37:二维列表去重[[1,2,3],[3,2,1],[1,2,4],[4,5,6],[2,1,4]]去重后得到[[1,2,3],[1,2,4],[4,5,6]]

temp\_set = set()

for li in arr:

li.sort()

temp\_set.add(tuple(li))

[list(i) for i in temp\_set]

38：说明简单下MVT和MVC的区别，及Django的特点。

MVT 来自于 MVC 的思想; M-M; V-C; T-V 对应关系

Django是当下比较流行一个基于python开发的web框架,它功能多样,扩展性强

其中有两个核心概念: 视图 和 序列化器

基于Django 结合 restful 风格二次开发了 drf 框架. 实际开发中应用此框架更广.

39：下面需求使用类视图完成。

1. 在Django中使用MySQL数据库，创建meiduo数据库，

2. 定义商品的模型类，商品字段包含：名字、价格、评论量、销售量、描述信息这几个字段。

3. 完成数据迁移，

4. 实现增加商品的功能。

5. 把增加的商品信息以json形式返回。

40：DRF 中的Request和Response与Django中的HttpRequest、HttpResponse有什么区别。