

CRM、权限、Linux 考试

一、Python 基础 (13 分)

1. 请简述赋值、浅拷贝、深拷贝的区别。(3 分)

赋值 (=), 就是创建了对象的一个新的引用, 修改其中任意一个变量都会影响到另一个。

浅拷贝: 创建一个新的对象, 拷贝了原始对象中包含项的引用 (如果用引用的方式修改其中一个对象, 另外一个也会改变)

深拷贝: 创建一个新的对象, 并且递归的复制它所包含的对象 (修改其中一个, 另外一个不会改变)

2. 简述__new__和__init__的区别。(3 分)

当创建一个新实例时调用__new__, 初始化一个实例时用__init__。

__new__是一个静态方法, 而__init__是一个实例方法。

__new__方法会返回一个创建的实例, 而__init__什么都不返回。

只有在__new__返回一个 cls 的实例时后面的__init__才能被调用。

3. 装饰器的作用是什么? 请写出一个打印函数运行时间的装饰器。(3 分)

装饰器本质上是一个 Python 函数, 它可以让其他函数在不需要做任何代码变动的前提下增加额外功能。

```
import time
```

```
def wrapper(func):
```

```
    def inner(*args, **kwargs):
```

```
        start = time.time()
```

```
        ret = func(*args, **kwargs)
```

```
        run_time = time.time() - start
```

```
        print(run_time)
```

```
        return ret
```

```
    return inner
```

```
@wrapper
```

```
def func():
```

```
    print("s13")
```

4. 请列举几种你经常用的 Python 包, 并解释其功能以及用法。(2 分)

django 常用的 web 框架

pymysql 连接操作 MySQL 数据库

hashlib 加密

request 爬虫发送请求

5. 一句话打印出 0-100 内的偶数。(2 分)

```
print([i for i in range(101) if i%2==0])
```

二、CRM (30 分)

1. CRM 系统是提供给谁使用的? 都为不同用户提供了哪些功能 (业务)? (4 分)

CRM 系统是提供给销售、班主任、财务等人员使用的。

销售功能:

录入客户相关信息、记录跟进客户情况、添加报名表、添加缴费记录及相关内容的展示

班主任功能:

创建编辑班级信息、创建课程记录、批量生成学生学习记录及相关内容的展示

财务功能:

财务审批等

2. 什么是公户? 什么是私户? 为什么要做这个区分? (3 分)

在 CRM 系统中, 将没有绑定销售的客户称为公户, 绑定销售的客户称为私户。

销售人员各位维护自己的客户(私户), 不可以查看或者跟进其他人的客户, 避免产生纠纷。

3. 请列举出 CRM 系统中的表。(5 分)

4. 请详细说说你们公司销售是如何使用 CRM 的。(3 分)

销售可以通过个人渠道获得客户信息, 自己录入到系统中, 或者销售主管或网络咨询师将已录入系统的客户分配给销售。

销售需要在规定时间内对客户进行跟进或回访, 将跟进信息记录到系统中, 根据不同的情况标记客户的报名意向。

当客户确定报名时需要让客户本人或者销售帮忙添加报名表。

当销售受到客户缴纳的费用时, 需要填写缴费记录, 并将费用转交财务进行审批。

5. CRM 中有哪些技术点, 用于做什么(至少写出 5 个)? (5 分)

auth 模块 认证

modelform 对表进行增删改查

分页 对数据进行分页显示

Q 对象 查询条件或的关系

QueryDict 保留搜索条件

url 的命名和反向解析

事务+行级锁 加入私户

批量创建对象

modelformset

三、权限 (32 分)

1. 使用几张表来完成权限控制的? 每张表中有哪些字段? (5 分)

菜单表 (Menu)

id name

权限表 (Permission)

id title url name parent_id menu_id

角色表 (Role)

id name

用户表 (User)

id username password

角色和权限关系表

id role_id permission_id

用户和角色关系表

id user_id role_id

2. 请简述实现权限控制的流程。（6 分）

在 web 应用中，一个 url 代表一个权限，可访问某一 url 则说明有该权限。该权限系统中 URL 被分类成两种，一种是可以做二级菜单的父权限，一种是不能做菜单的子权限。Menu 表中的数据称之为一级菜单，二级菜单被分配给一级菜单。子权限分配给父权限。

当用户登录成功时，根据用户所有的角色查询出所具有的权限，将权限信息和菜单信息存放在 session 中。登录成功后跳转至其他 URL 时，请求经过中间件时进行权限的校验，根据当前访问的 url 和 session 中存放的权限信息进行正则匹配。都匹配不成，则拒绝访问。匹配成功则继续走正常的流程得到相应的响应。

页面中的可点击的按钮也是权限，在模板渲染时使用自定义 filter 判断该按钮所代表的权限是否在该用户所拥有的权限中，如果是则显示按钮，否则不显示。

3. 初始化权限时，定义了两个字典，分别是什么数据结构？为什么要把他们存在 session 中？（6 分）

存放权限信息的字典：

```
permission_dict={
    "权限的 URL 别名":{
        "url":"url 路径",
        "id":"权限的 id",
        "pid":"父权限的 id",
        "title":"权限的标题",
        "pname":"父权限的 URL 别名"
    }
}

menu_dict = { 2: {
    'title': '客户管理',
    'icon': 'fa-address-book-o',
    'children': [{
        'title': '客户列表',
        'url': '/customer/list/',
        'id': 1
    }]
},
}
```

将权限和菜单的字典存放在 session 中是为了给不同用户存放不同的权限和菜单，并且 session 可以存放在 Redis 中，读取速度快。

4. 权限校验的中间中都做了什么事情？（4 分）

1. 获取当前访问的 URL，判断是否在不需要权限校验的白名单中，如果在则不再进行校验。
2. 从 session 中获取到该用户的权限信息。
3. 循环出每一个权限进行正则匹配，匹配成功则改变 flag 标志位为 True，并且获取到需要默认展开菜单的权限 id 和要展示的路径导航信息。
4. 循环结束后，Flag 还为 False 则拒绝访问。

5. 是如何实现二级菜单默认展开并且是选中状态的？（5 分）

二级菜单是通过 inclusion_tag 实现的，将 request 当做参数传给 menu 函数，函

数中从 request 中获取到 menu_dict。循环 menu_dict 给一级菜单都添加了一个 hide 的 class，默认一级菜单下的二级菜单都是隐藏的。再循环一级菜单下的二级菜单，判断当前的二级菜单（父权限）是否是要默认选中的二级菜单，如果是的话，给该二级菜单添加一个 active 的 class，给该二级菜单对应的一级菜单的 class 重新赋值为空。

模板中生成二级菜单时添加上对应的 class，则显示出相应的状态。class 为 hide 则是隐藏状态，class 为 active 则是选中状态。

6. 权限批量操作时，哪些权限是需要新增的？哪些是需要删除的？哪些是可更新的？（3 分）
7. 开发过程中遇到问题，一般都是怎么解决的？请说一下你印象比较深刻的一个问题。（3 分）

四、Linux（25 分）

1. 请写出操作文件/文件夹的基本命令（3 分）

文件

cat,touch,vim,ls,mv,rm

文件夹

cd,mkdir,rm,mv,ls

2. 找到目录/opt/下所有的以.py 结尾的文件（1 分）

find /opt -name *.py

3. 使用 grep 过滤出/etc/passwd 下，除了/sbin/nologin 的信息，且打印行号（1 分）

grep -vn "/sbin/nologin" /etc/passwd

4. 不间断打印/var/log/django/message.log 的日志信息（1 分）

tail -f /var/log/django/message.log

5. linux 文件权限的 755,700 是什么意思？转化为 rwx 格式是什么？（2 分）

rwxr-xr-x 是 755 代表文件属主可读可写可执行，属组是可读可执行，其他人是可读可执行

rwx----- 是 700 代表文件属主可读可写可执行

6. 解压缩 Python 源码包 Python-3.7.tgz.gz（1 分）

tar -zxvf Python-3.7.tgz.gz

7. 在/opt/目录下创建 django 项目 mycrm 的命令是？（1 分）

cd /opt/

django-admin startproject mycrm

8. 在/opt/mycrm 中创建 app01 的命令是？（1 分）

cd /opt/mycrm

django-admin startapp app01

9. 如何创建/usr/bin/python3 软连接，链接到/opt/python34/bin/python3（2 分）

ln -s /opt/python34/bin/python3 /usr/bin/python3

10. nginx 是什么？（1 分）

nginx 是高性能的 web 服务器

11. 每月的 5,15,25 天的晚上 5 点 50 重启 nginx（2 分）

50 17 5,15,25 * * /usr/bin/systemctl restart nginx

12. 编译安装 nginx 有几个步骤？（3 分）

1. 下载 nginx 源码包

2. 解压缩 nginx 源码包

3. `configure --prefix=/opt/nginx/` 释放编译文件，指定 nginx 安装目录
 4. `make && make install` 编译和编译安装
13. 请简述发布 uWSGI+Django+virtualenv+supervisor+nginx 的过程，（简述安装配置部署过程）（6 分）
1. 准备好 python3 和 virtualenv 环境
 2. 激活虚拟环境，在虚拟环境中创建 django 项目
 3. 在虚拟环境中安装 uWSGI 模块
 4. 配置 `nginx.conf`，使得 nginx 反向代理 uWSGI，启动 nginx
 5. 安装 supervisor 工具，在配置文件中写好启动 uWSGI+django 的命令
 6. 启动 supervisor，同时启动了 uWSGI+django
 7. 访问 10.0.0.10 的 80 端口，查看 nginx 反向是否生效，访问到 django 项目

内部资料 禁止外泄