基于python的web开发

笔记地址

---

1. 知识点和实际开发项目结合
2. 课后留作业,下节课看完成情况
3. 每位同学所掌握的知识层次不齐,需考虑到
4. 课后接收同学发骢,下节课进行调整,
   包括课程内容,课程难易程度,课程时长等

准备篇(2021年12月27日)

一台能上网的笔记本(工具兼查资料)

linux的安装(ubuntu,centos,kali)

python的安装(python3)

编辑器(vi,vscode,sublime等)

笔记(md,txt,word)

linux的安装

直接装在计算机上,虚拟机,子系统(WSL)

直接ubuntu安装

虚拟机安装

子系统安装

注意的点:

- 更新软件源
- 各linux系统命令会有所差异

  apt,yum

python的安装

centos安装python3

更新pip源

也可以直接用命令

```
pip config set global.index--url https://pypi.tuna.tsinghua.edu.cn/simple
```

编辑器

Vim

Vscode

Sublime

Pycharm

notepad++

...

笔记

typora

markdown

```python
def test():
    print('这是一段python代码')
```

Linux
MacOS
Windows

2021年12月27日作业

作业

在自己的电脑上,用虚拟机装一个centos7.9或者ubuntu18.04

更新软件源为阿里源或者清华源

启动之后,安装python3(ubuntu18.04默认装了3.6)

1. 编写 一个 xxx.py 文件，写入Python 的语句

```
# file: hello.py
print("hello world!")
```

2. 用 python3 的解释执行器执行 xxx.py
    - 在Windows 用终端命令

```
C:\> python hello.py
```

    - 在Linux 下用终端命令

```
$ python3 hello.py
```

预习

linux有哪些命令

2021年12月27日作业情况

1.root用户操作

2.没有虚拟环境

linux(2022年1月4日-1)

常用命令

command [-options] [parameter]

开始

ls,cd,pwd,whoami

建用户

如：zbxu，请使用root用户操作（root用户请询问银行管理员）

```
useradd -m zbxu
```

设置密码

```
passwd zbxu
```

sudo权限

```
usermod -a -G wheel zbxu
```

增删改

mkdir,rm,mv,cp,touch,rm,vi

查

find,grep,cat,more,less,tail,head,ps ux,↑↓

管道

ps ux | grep sh | grep -v grep

权限管理

sudo 管理员权限执行

chmod +[o]x xxx.txt

chmod 777 xxx.txt

例如：
`777` ===> `u=rwx,g=rwx,o=rwx`
`755` ===> `u=rwx,g=rx,o=rx`
`644` ===> `u=rw,g=r,o=r`

解压缩

tar -zcvf xxx.tar.gz 要打包的文件

tar -zxvf xxx.tar.gz

zip xxx.zip 要打包的文件

unzip xxx.zip

重定向

ls -la >> ~/xxx.log

符号链接

ln -s hello.py hello

关机重启

shutdown -r now 立即重启

shutdown now 立即关机

shutdown +10 10分钟后关机

shutdown -c 取消关机计划

帮助

xxx --help

man xxx

shell编程

启动停止脚本

```
#!/bin/sh
filetime=`date +%Y%m%d`
case "$@" in
    start)
        nohup python hello.py >> ~/logs/hello.log &
        ;;
    stop)
        ps ux|grep hello|grep -v grep|awk '{print $2}'|xargs kill -9
        ;;
    *)
        echo "unknown argument args(start|stop)"
        exit 1
        ;;
esac
```

vim 粘贴

```
set paste
```

命令分解

重复执行脚本

```
import time
print('hello', time.time())
```

```
#!/bin/sh
while true
    do
        python time_print.py
        echo "wait 5s"
        sleep 5
    done
```

虚拟环境(2022年1月4日-2)

目的

相互隔离的 Python 环境

指的说明的是包管理器除了pip还有其他,遇到了不要感觉陌生

相关概念链接

python 包管理器

软件包源中的软件包数量巨大，版本多样，所以需要借助于软件源管理工具，例如 pip、conda、Pipenv、Poetry 等

- pip 是最常用的包管理工具，通过 `pip install <packagename>` 命令格式来安装软件包，使用的是 pypi 软件包源
- conda 多用作科学计算领域的包管理工具，功能丰富且强大，使用的软件包源是 Anaconda repository 和 Anaconda Cloud，conda 不仅支持 Python 软件包，还可以安装 C、C++ 、R 以及其他语言的二定制软件包。除了软件包管理外，还能提供相互隔离的软件环境。
- Pipenv 是 Kenneth Reitz 在2017年1月发布的Python依赖管理工具，现在由PyPA维护。Pipenv 会自动帮你管理虚拟环境和依赖文件，并且提供了一系列命令和选项来帮助你实现各种依赖和环境管理相关的操作
- Poetry 和 Pipenv 类似，是一个 Python 虚拟环境和依赖管理工具，另外它还提供了包管理功能，比如打包和发布。你可以把它看做是 Pipenv 和 Flit 这些工具的超集。它可以让你用 Poetry 来同时管理 Python 库和 Python 程序

很多包管理工具不仅提供了基本的包管理功能，还提供了虚拟环境构建，程序管理的等功能

创建

virtualenv

在 python3.3 之前，只能通过 virtualenv 创建虚拟环境，首先需要安装 virtualenv

```
pip install virtualenv
```

指定python解析器

```
virtualenv --python=python3 myenv
```

virtualenv: error: unrecognized arguments: --no-site-packages

venv

Python3.3 之后，可以用模块 venv 代替 virtualenv 工具，好处是不用单独安装，3.3 及之后的版本，都可以通过安装好的 Python 来创建虚拟环境:

```
python -m venv myvenv
```

可以在当前目录创建一个名为 myvenv 的虚拟环境

因为 venv 是依附于一个 Python 解析器创建的，所以不需要指定 Python 解释器版本

激活

```
$ source myvenv/bin/activate
```

激活后，可以在命令行中看到虚拟环境标记

判断当前虚拟环境的python版本:

```
which python
```

退出

```
deactivate
```

部署

之所以在开发时选择虚拟环境，除了避免库之间的冲突，还有重要的原因是方便部署，因为虚拟环境是独立的，仅包含了项目相关的依赖库，所以部署的效率更高，风险更小

一般部署流程是:

1. 开发完成后，使用 `pip freeze > requirements.txt` 命令将项目的库依赖导出，作为代码的一部分
2. 将代码上传到服务器
3. 在服务器上创建一个虚拟环境
4. 激活虚拟环境，执行 `pip install -r requirements.txt`，安装项目依赖

ssh(2022年1月4日-3)

口令

```
ssh user@host
```

需要输入密码

公钥

创建

```
ssh-keygen
```

一路回车即可

在~/.ssh中有生成的密钥:

id_rsa

以及公钥:

id_rsa.pub

将公钥的内容拷到对方服务器中的authorized_keys中,若原先已有内容,往后追加

如自己新建的authorized_keys,则需要配置文件权限

文件权限

加完记得重启ssh服务

查看服务状态

```
systemctl status sshd
```

重启服务

```
systemctl restart sshd
```

启动服务

```
systemctl start sshd
```

停止服务

```
systemctl stop sshd
```

使用

```
ssh user@host
```

不用输入密码

2022年1月4日作业

作业1

1. 在自己创建的用户的家目录创建三个目录,分别是scripts,logs,projects
2. 在scripts中写一个sh脚本(print_time.sh),
3. 启动print_time.sh脚本,每隔30s输出当前时间,格式如下:

   2022年1月4日 10:43:22

   并且重定向到~/logs/homework.log中(不需要输出到前台)

作业2

1. 在~/projects/中创建一个python3的虚拟环境py3
2. 进入虚拟环境
3. 在环境中使用pip安装flask
4. 启动一个web服务,浏览器打开

   服务器ip:8888/自己的名字

   能够看到hello 自己的名字
5. 退出虚拟环境

作业3

1. 使用公钥ssh连上自己的虚拟机,启动flaskweb服务
2. 主机(非虚拟机)的浏览器打开网页显示

   hello 自己的名字

预习

python 的基础语法

python中包的使用

python

语法

web框架

django flask

爬虫

requests post get

数据库

oracle mysql postgresql

nginx,apache

反向代理

git

代码管理