

任务 1. 立项

内 容：明确实训要求和纪律，组建团队，确定分工(组长、开发、测试)

学 时：2 学时

1. 实训的要求和纪律

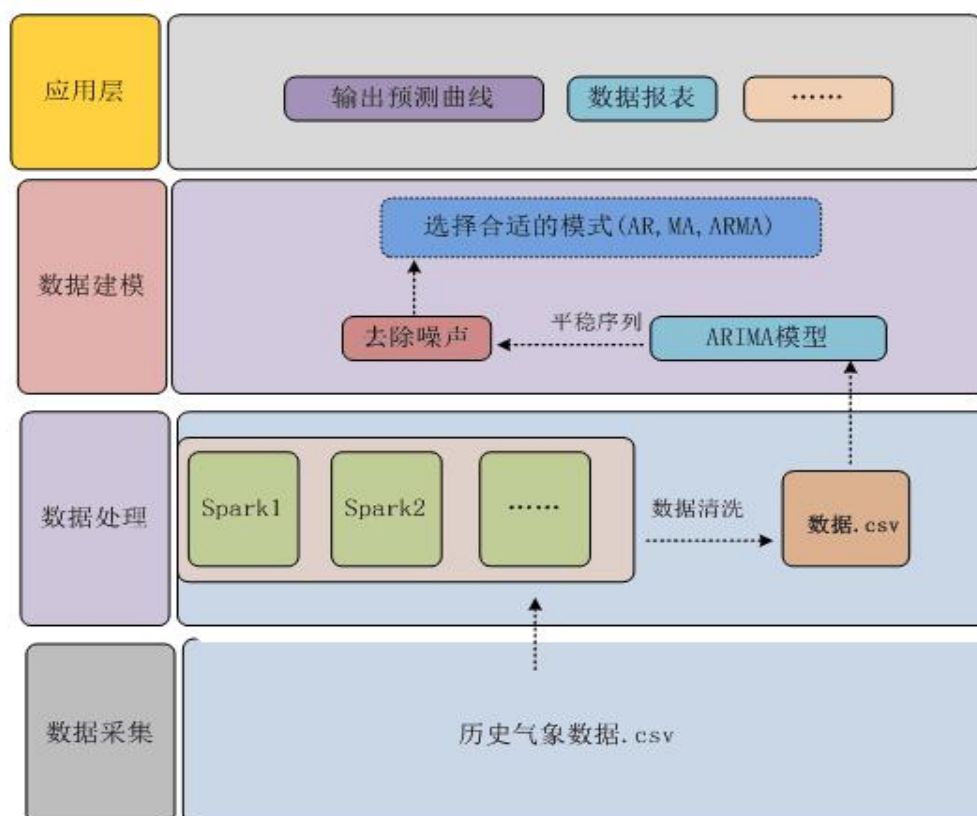
- 第一周结束后进行一次中期答辩，内容包括阶段成果展示和下届阶段任务安排。
- 实训结束后进行项目答辩、成绩评定。
- 每天进行小组早站立会，老师和组长进行站立会。
- 成绩根据项目成果展示的情况和本阶段每天的任务进展情况构成，每天的任务进展通过平台上进行跟踪。
- 每天早上清点任务列表和负责人，每天晚上总结任务进度和需要协调的资源。

2. 实训内容

本次实训要求完成一个基于天气历史数据的分析预测前端程序和后台服务两部分，需要提交的材料包括：项目文档，项目代码，答辩 PPT。

本次实训项目是基于时间序列模型对历史气象数据进行分析，数据采集 NCDC 气象官方网站。主要功能是对北京地区 1980 年至 2020 年的气象数据进行 ETL 清洗后，使用 spark 对数据进行筛选，筛选出某年中每一天的最低气温，最高气温和平均气温，然后使用 Python 对数据进行时间序列分析，预测一周的气温走势。最后，使用 web 终端对分析的结果进行可视化展示。

系统架构图如下所示：



任务指导

- 1) 确定分组和小组名称，分组原则：技术力量平衡（组员要各有特长，合理搭配，涉及技术，UI，文档，测试，所以分组时技术好的同学都在一组未必是好事），男女平衡
- 2) 登录云平台，在云平台上创建项目，为了避免项目重名，项目名称为项目名（+组长名），加进组员和老师，
- 3) 确定 2 次迭代周期的时间，每 5 天一个迭代周期

任务实现

无

任务 2. Python 环境安装配置

任务描述

内 容：安装配置开发所需的 python 环境

学 时： 2 学时

知识点：配置 Python 运行环境

重 点：

难 点： python 知识

本项目用到的第三方库比较多，如果每台机器都要手动安装第三方任务包，比较费时间，可以把一套配置好的 python 目录打包复制到学生机器上即可。

任务指导

1. 安装配置 python
2. 设置 python 运行环境

任务实现

- 1、下载 python

Python 最新源码，二进制文档，新闻资讯等可以在 Python 的官网查看到：

Python 官网：<https://www.python.org/>

你可以在以下链接中下载 Python 的文档，你可以下载 HTML、PDF 等格式的文档。Python 文档下载地址：<https://www.python.org/doc/>

Python 已经被移植在许多平台上（经过改动使它能够工作在不同平台上）。

您需要下载适用于您使用平台的二进制代码，然后安装 Python。

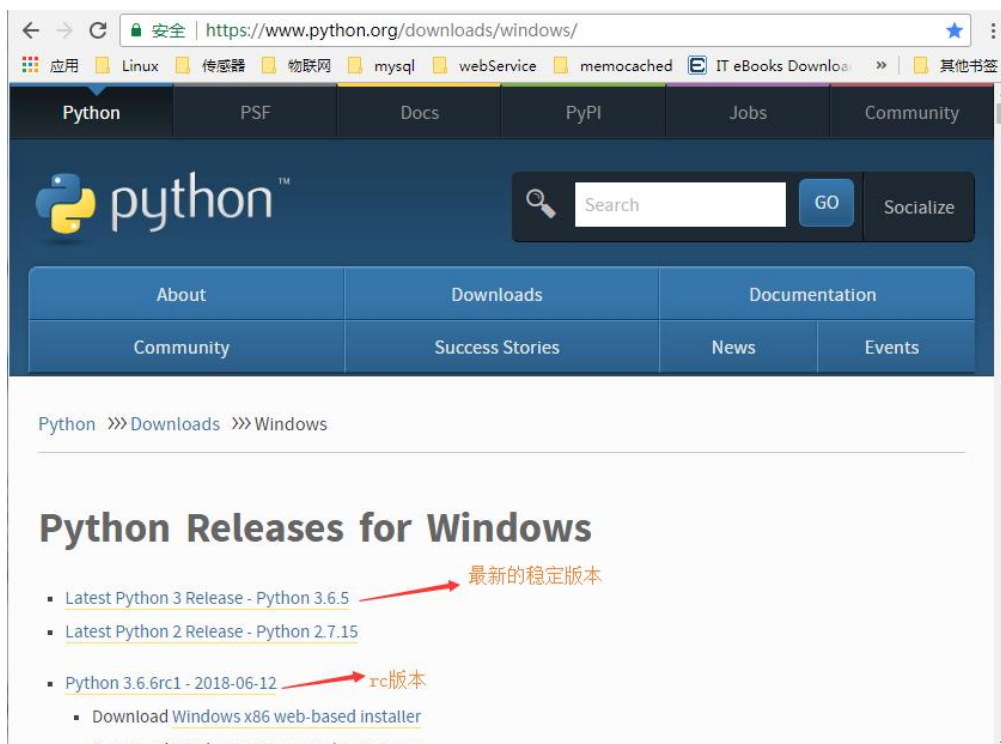
如果您平台的二进制代码是不可用的，你需要使用 C 编译器手动编译源代码。

编译的源代码，功能上有更多的选择性，为 python 安装提供了更多的灵活性。

以下是各个平台安装包的下载地址：



我们选择 Window 平台安装 Python，单击“windows”选项，出现如下画面：



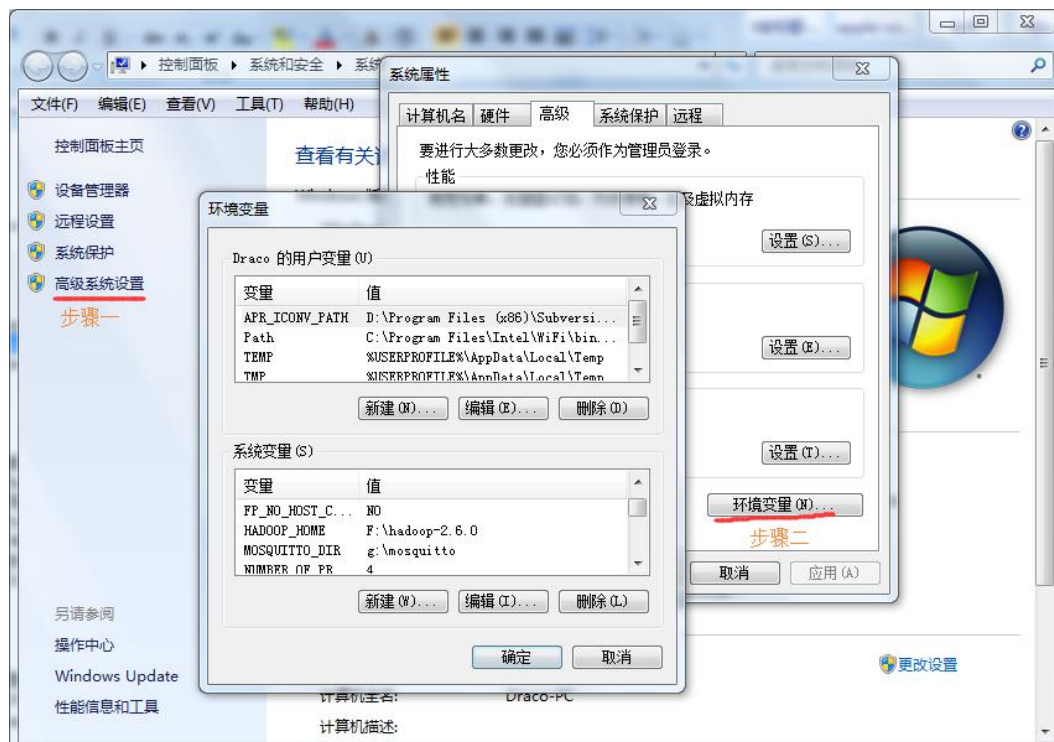
在下载列表中选择 Window 平台安装包,包格式为:python-XYZ.msi 文件 , XYZ 为你要安装的版本号, 我们选择 python3.6.5, 点击出现下一页面:

Files			
Version	Operating System	Description	MD5 Sum
Gzipped source tarball	Source release		ab25d24b1f8cc4990ade979f6dc37883
XZ compressed source tarball	Source release		9f49654a4d6f733ff3284ab9d227e9fd
macOS 64-bit/32-bit installer	Mac OS X	for Mac OS X 10.6 and later	bf319337bc68b52fc7d227dca5b6f2f6
macOS 64-bit installer	Mac OS X	for OS X 10.9 and later	37d891988b6aeedd7f03a70171a8420d
Windows help file	Windows		be70202d483c0b7291a666ec66539784
Windows x86-64 embeddable zip file	Windows	for AMD64/EM64T/x64	04cc4f6f6a14ba74f6ae1a8b685ec471
Windows x86-64 executable installer	Windows	for AMD64/EM64T/x64	9e96c934f5d16399f860812b4ac7002b
Windows x86-64 web-based installer	Windows	for AMD64/EM64T/x64	640736a3894022d30f7babff77391d6b
Windows x86 embeddable zip file	Windows		b0b099a4fa479fb37880c15f2b2f4f34
Windows x86 executable installer	Windows		2bb6ad2ecca6088171ef923bca483f02
Windows x86 web-based installer	Windows		596667cb91a9fb20e6f4f153f3a213a5

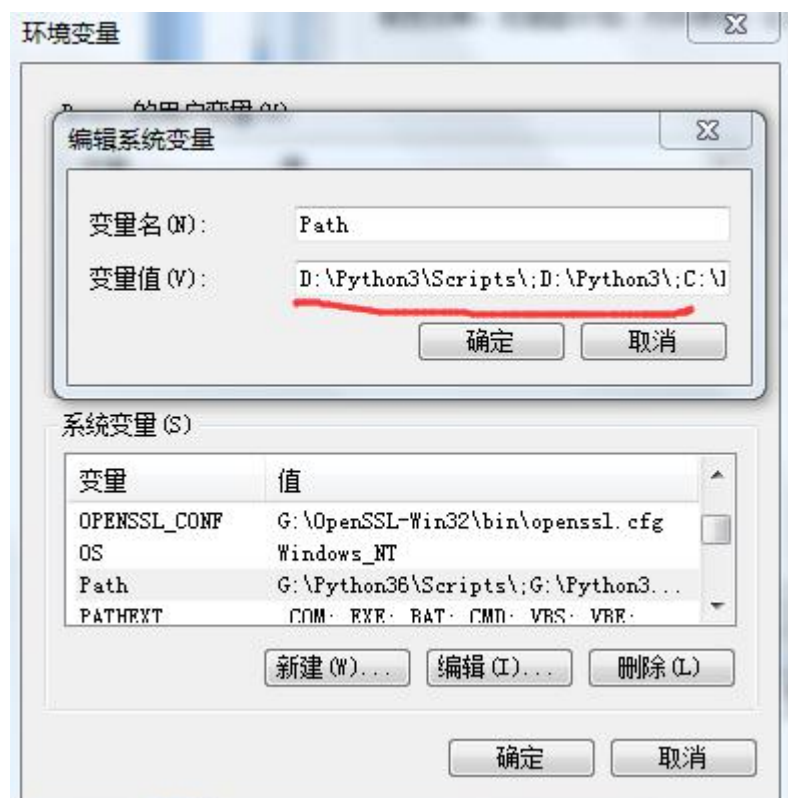
点击对应电脑的版本，开始下载，下载完成后双击安装即可，

2、配置 python

安装完成后，假设安装目录是 D:\python3 目录，则首先需要配置环境变量以 windows 7 系统为例，在“计算机”图标上右键，属性，点“高级系统设置”再点“环境变量”出现如下画面：

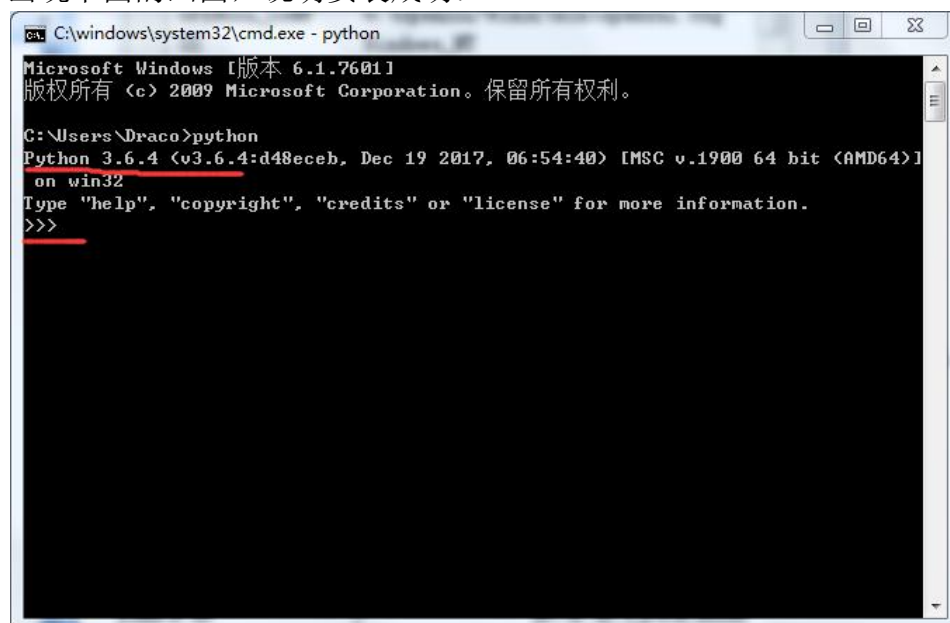


在系统变量栏内，找到 path 变量值，添加上
D:\Python3\Scripts\;D:\Python3\



点确定即可。

要验证安装的 python 是否正常工作，可以打开命令行窗口，输入 python，如果出现下面的画面，说明安装成功：



任务 3. PyCharm 安装配置

任务描述

内 容：安装配置开发运行环境 pycharm
学 时： 2 学时
知识点： 开发配置 pycharm、配置 Python 运行环境
重 点： pycharm 介绍
难 点： python 知识

学生根据“任务实现”完成任务内容，

任务指导

1. 安装配置 pycharm

任务实现

- 1、安装配置 pycharm

首先去 PyCharm 官网, 或者直接在浏览器中输入网址:

<https://www.jetbrains.com/pycharm/download/#section=windows>

Download PyCharm

Windows

macOS

Linux

Professional

Full-featured IDE
for Python & Web
development

DOWNLOAD

Free trial

Community

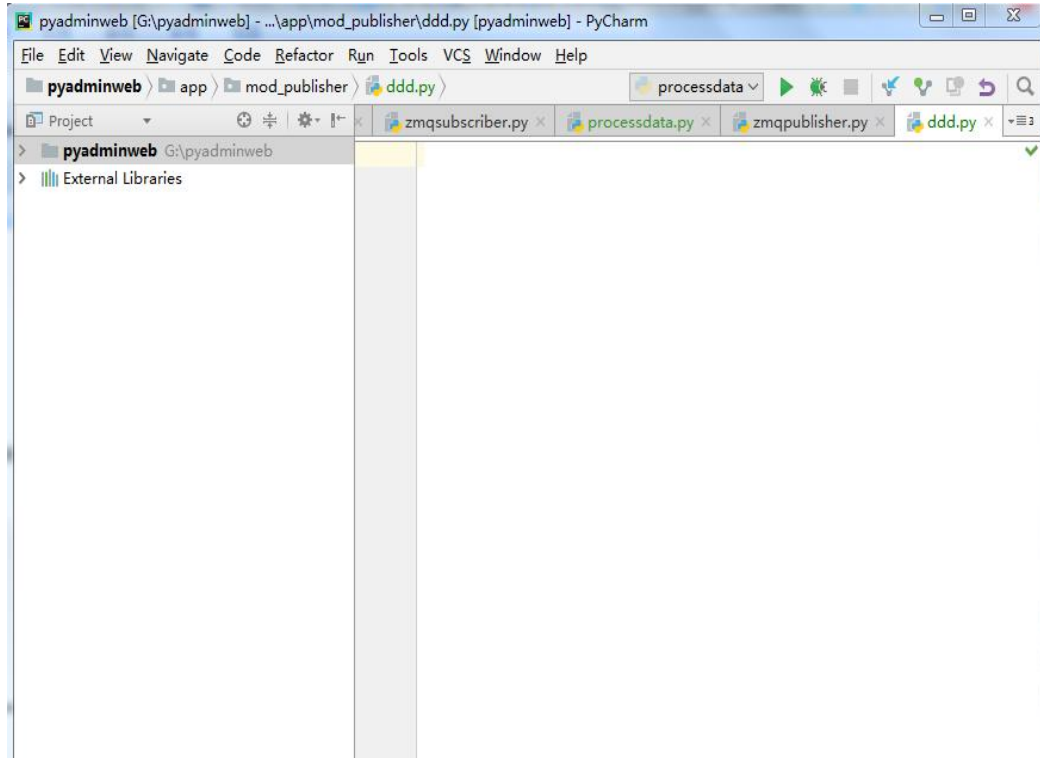
Lightweight IDE
for Python & Scientific
development

DOWNLOAD

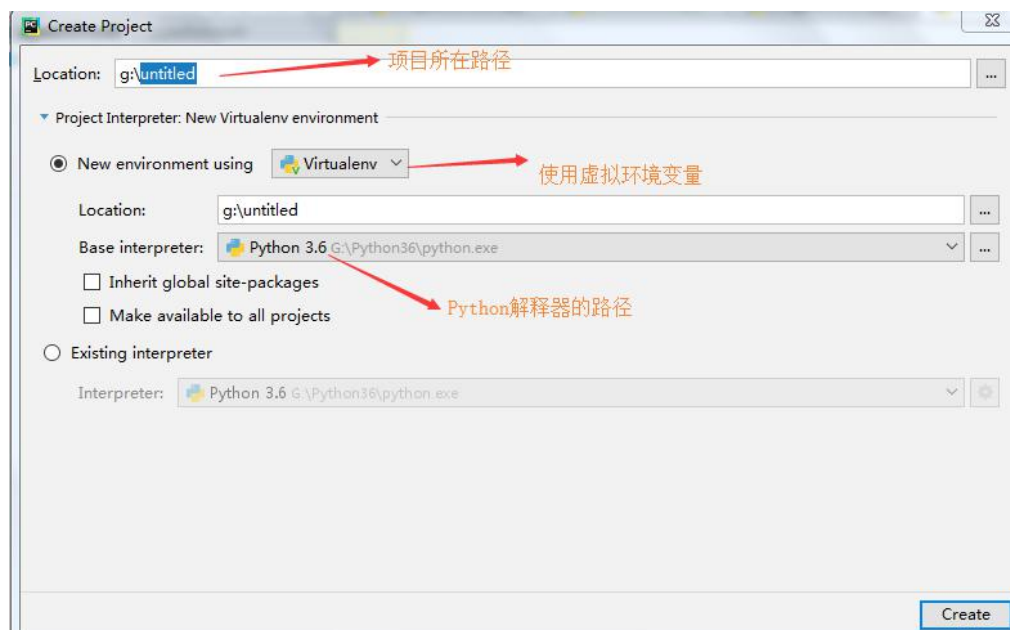
Free, open-source

选择不同的操作系统，然后选择 Community 版本, 点 Download，下载完成后, 双击安装包开始安装程序。

安装完成后，双击打开，如下所示：



在菜单栏，点 File，选 new project



其中 location 是项目的存储目录，

项目编译器可以选择 virtualenv（虚拟环境）：

virtualenv 是一个创建独立 Python 运行环境的工具，主要解决 Python 软件开发过程中版本和依赖性问题。比如，有一个应用需要用到 matlab1.0，但其他应用要用 2.0。那如何解决应用间兼容问题？比如在一个共享主机上，不能直接将包安装在全局目录下的问题。virtualenv 创建一个拥有自己安装目录的环境，各个环境之间的库各自独立。

也可以不使用虚拟环境，直接在原有的 python 运行环境中运行（第二个选项 existing interpreter）

如果前面已经配置了 python 运行环境，则 Interpreter 框里就会出现 python 的路径，选择即可。

我们选择 virtualenv，存储路径输入“pyadminweb”，点“create”，新项目创建成功。

任务 4. Java 环境配置

任务描述

内 容：介绍 Java 环境的安装配置

学 时： 2 学时

知识点：

重 点：

难 点：

Java 环境是 Spark 的依赖环境，必须提前安装配置

任务指导

1. 下载安装 JDK1.8
2. 配置 java 运行环境

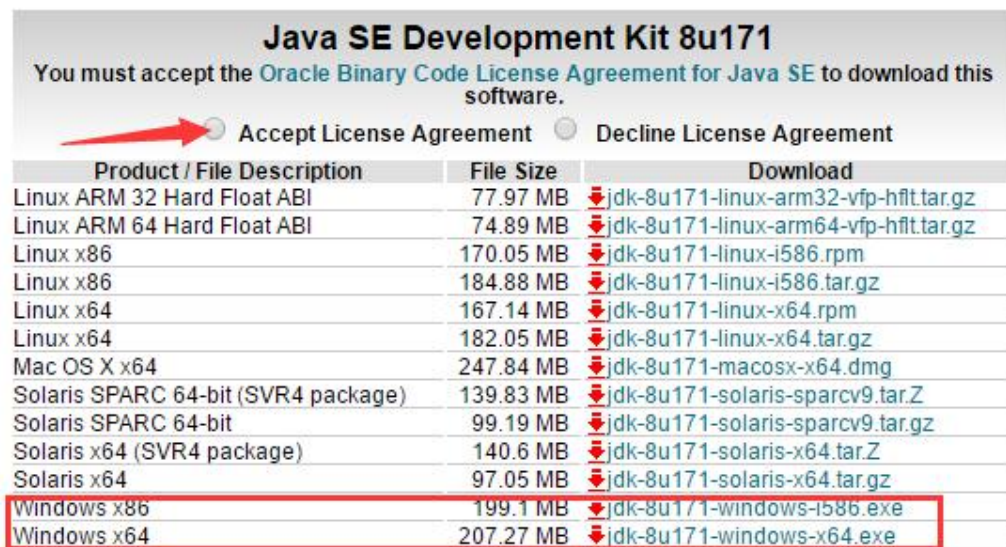
任务实现

1、下载 jdk1.8

下载网址：

<http://www.oracle.com/technetwork/java/javase/downloads/jdk8-downloads-2133151.html>

选择 32 位或者 64 位 window 版本，点击下载



2、安装 jdk1.8 并配置环境变量

单击“计算机-属性-高级系统设置”，单击“环境变量”，在“系统变量”栏下单击“新建”，创建新的系统环境变量。

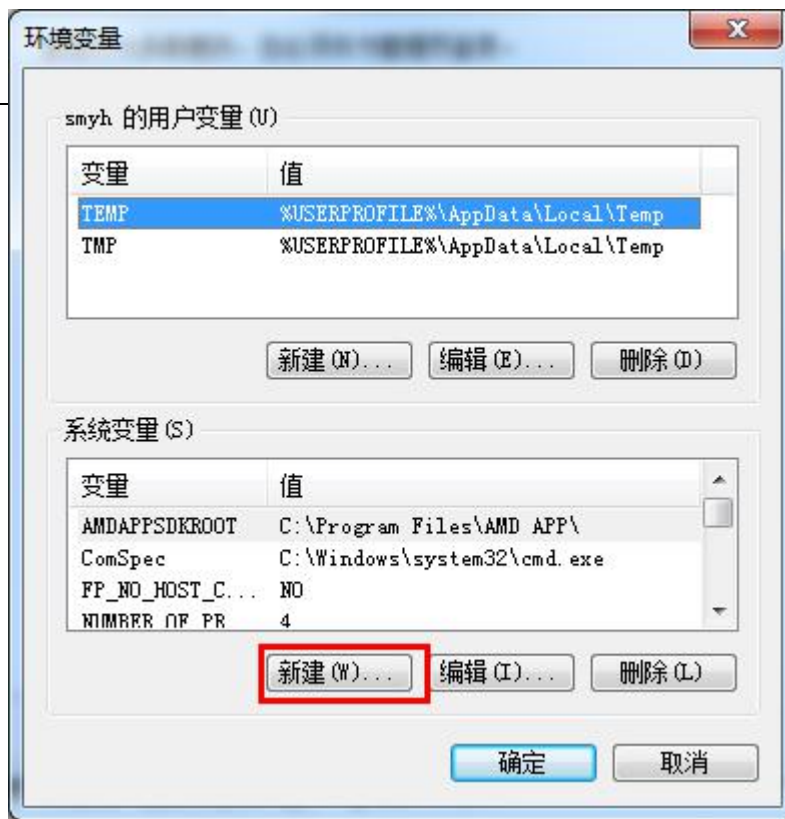
(1)新建变量名"JAVA_HOME", 变量值"C:\Java\jdk1.8.0_05"（即 JDK 的安装路径）

(2)编辑变量名"Path", 在原变量值的最后面加上“;%JAVA_HOME%\bin;%JAVA_HOME%\jre\bin”

(3)新建变量名“CLASSPATH”,变量值

“.;%JAVA_HOME%\lib;%JAVA_HOME%\lib\dt.jar;%JAVA_HOME%\lib\tools.jar”

如：JAVA_HOME 环境变量的操作如下：



2. 查看 jdk 是否正确安装

打开 windows 命令行， 输入 `java -version` 查看是否正确安装。

```
C:\Users\smyh>java -version
java version "1.8.0_05"
Java(TM) SE Runtime Environment (build 1.8.0_05-b13)
Java HotSpot(TM) Client VM (build 25.5-b02, mixed mode)

C:\Users\smyh>
```

任务 5. Spark 环境配置

任务描述

内 容：介绍 Spark 框架的安装配置

学 时： 2 学时

知识点： Spark 原理和相关应用

重 点：

难 点：

讲授学生 Spark 的基本原理和运行方式

任务指导

- 1, 下载 spark
- 2, 下载 hadoop
- 3, 配置环境变量

任务实现

- 1、下载 spark

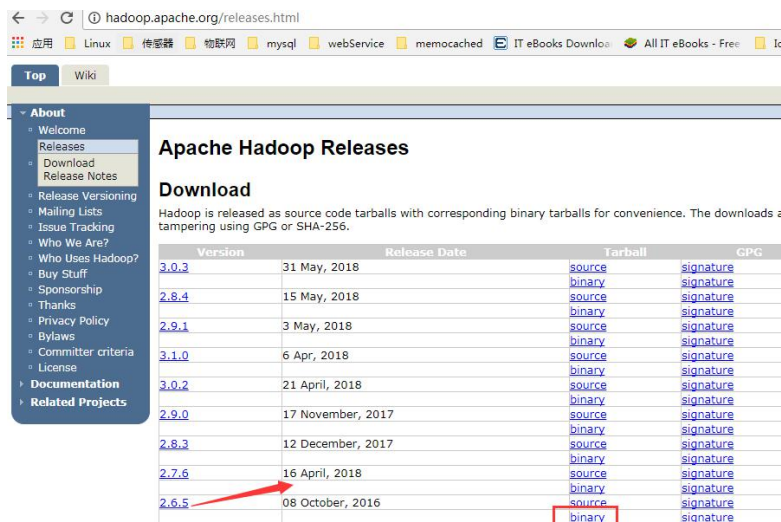
下载地址 <https://spark.apache.org/downloads.html>



下载完成后，解压缩至硬盘根目录，然后把文件夹名称修改为 spark

- 2、下载 hadoop

网址是：<http://hadoop.apache.org/releases.html>



选择 2.6.5 版本，点 binary 文件，进行下载
 下载完成后解压缩至 F 盘根目录，修改文件名为 hadoop

Windows 版本的用户，还需要额外的 hadoop 支持程序，
 文件名称是 hadoop2.6-bin.rar
 把里面的文件全部解压缩到 f:\hadoop\bin 目录下即可。

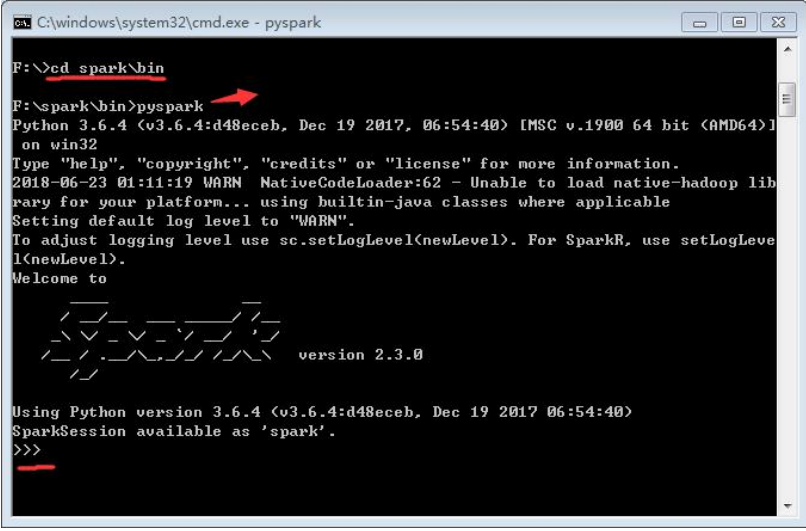
3、配置环境变量

首先配置 hadoop 的全局变量



新建系统变量 HADOOP_HOME=F:\hadoop
 在系统变量的 path 里加入 %HADOOP_HOME%\bin
 再添加一个 SPARK_HOME=F:\spark

全部设置完成后，打开一个命令行窗口，进入 F:\spark\bin 目录，然后输入 Pyspark, 如果出现如下界面，说明安装成功



```
C:\windows\system32\cmd.exe - pyspark
F:\>cd spark\bin
F:\spark\bin>pyspark
Python 3.6.4 (v3.6.4:d48eceb, Dec 19 2017, 06:54:40) [MSC v.1900 64 bit (AMD64)]
on win32
Type "help", "copyright", "credits" or "license" for more information.
2018-06-23 01:11:19 WARN NativeCodeLoader:62 - Unable to load native-hadoop lib
rary for your platform... using builtin-java classes where applicable
Setting default log level to "WARN".
To adjust logging level use sc.setLogLevel(newLevel). For SparkR, use setLogLeve
l(newLevel).
Welcome to

      /\_/\
     /__\ \
    /__\ \
   /__\ \
  /__\ \
 /__\ \
/_/\_/\
version 2.3.0

Using Python version 3.6.4 (v3.6.4:d48eceb, Dec 19 2017 06:54:40)
SparkSession available as 'spark'.
>>>
```

任务 6. 获取气象数据

任务描述

内 容： 获取原始的历史气象数据
学 时： 2 学时
知识点： 气象要素的相关知识
重 点：
难 点：

学生根据“任务实现”完成任务内容，

任务指导

任务实现

浏览器打开网址

<https://www.ncdc.noaa.gov/data-access/land-based-station-data>

Quick Links

- Land-Based Station
- Datasets
- Find a Station
- Station Metadata
- Climate Data Online
- Data Publications
- Satellite
- Radar
- Model
- Weather Balloon
- Marine / Ocean
- Paleoclimatology
- Severe Weather
- Blended & Global

Land-Based Station Data

Land-based observations are collected from instruments sited at locations on every continent. They include temperature, dew point, relative humidity, precipitation, wind speed and direction, visibility, atmospheric pressure, and types of weather occurrences such as hail, fog, and thunder. NCEI provides a broad level of service associated with land-based observations. These include data collection, quality control, archive, and removal of biases associated with factors such as urbanization and changes in instrumentation through time. Data on sub-hourly, hourly, daily, monthly, annual, and multiyear timescales are available.

- Datasets and Products**
Access NCEI's land-based datasets directly.
- Find a Station**
Locate a station by using either a map tool or a location and data search tool.

点” find a station”

■ Data Tools: Find a Station

Retrieve weather records from observing stations by ente can be specified as city, county, state, country, or ZIP code

Enter Location → 输入beijing, 选择北京市中国

beijing

- 📍 Beijing 北京市中国
- 📍 Beijing 中国
- 📍 Beijing South Railway Station 丰台区中国
- 📍 Beijing Road 越秀区广州市广东省中国
- 📍 Pekin 伊利诺伊州美国

powered by Google

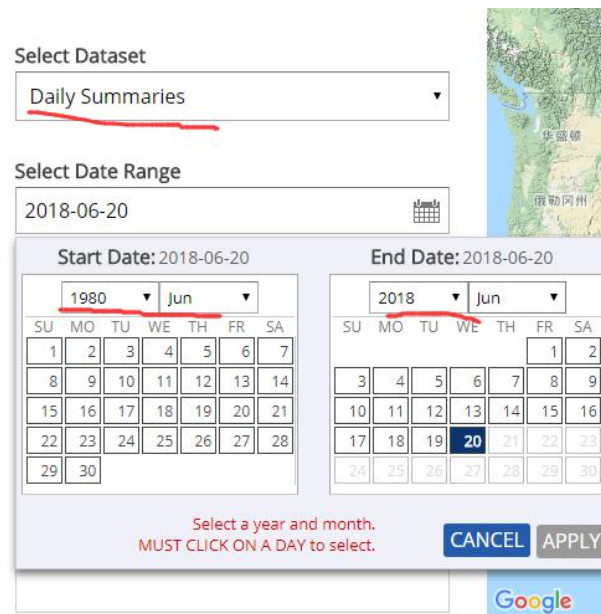
在 location 框里输入 beijing

选择”北京市中国”，

select dataset

选择 Daily Summaries

然后选择日期范围，尽量选择比较长的时间段，比如 1980 年 1 月 1 日到现在



Select Dataset

Daily Summaries

Select Date Range

2018-06-20

Start Date: 2018-06-20

End Date: 2018-06-20

1980 Jun

2018 Jun

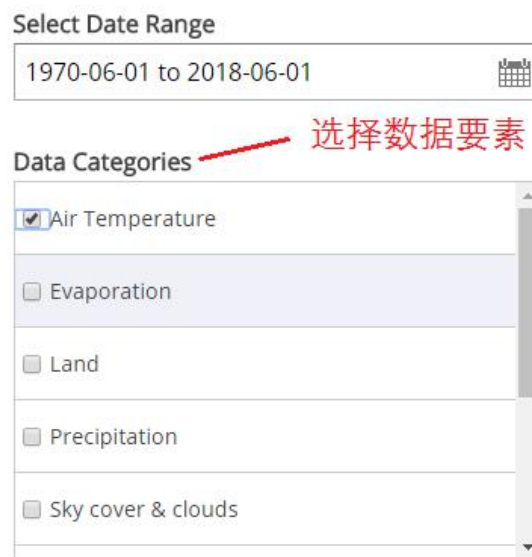
20

Select a year and month.
MUST CLICK ON A DAY to select.

CANCEL APPLY

Google

选择完时间和具体的日期后, 点” APPLY”
在接下来的画面中选择数据要素, 包括温度, 降水, 光照, 风力等



Select Date Range

1970-06-01 to 2018-06-01

Data Categories

☒ Air Temperature

☐ Evaporation

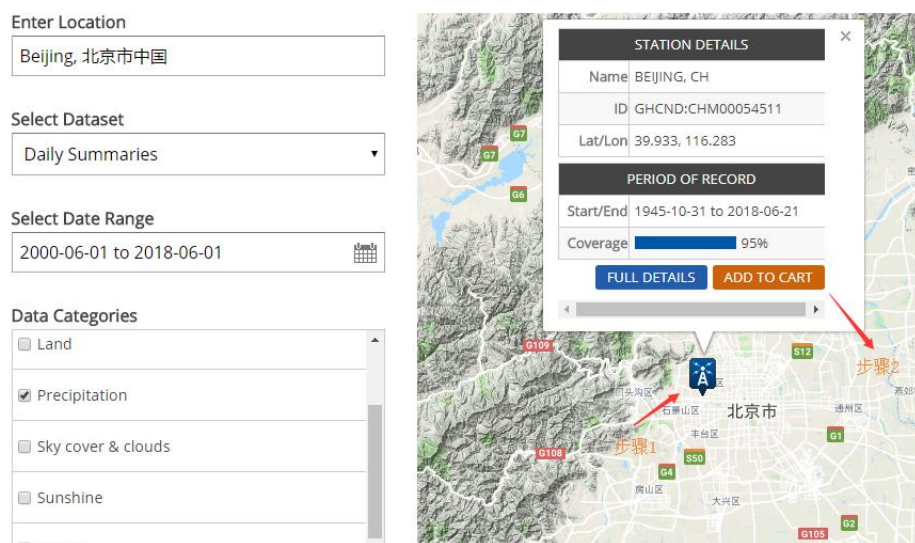
☐ Land

☐ Precipitation

☐ Sky cover & clouds

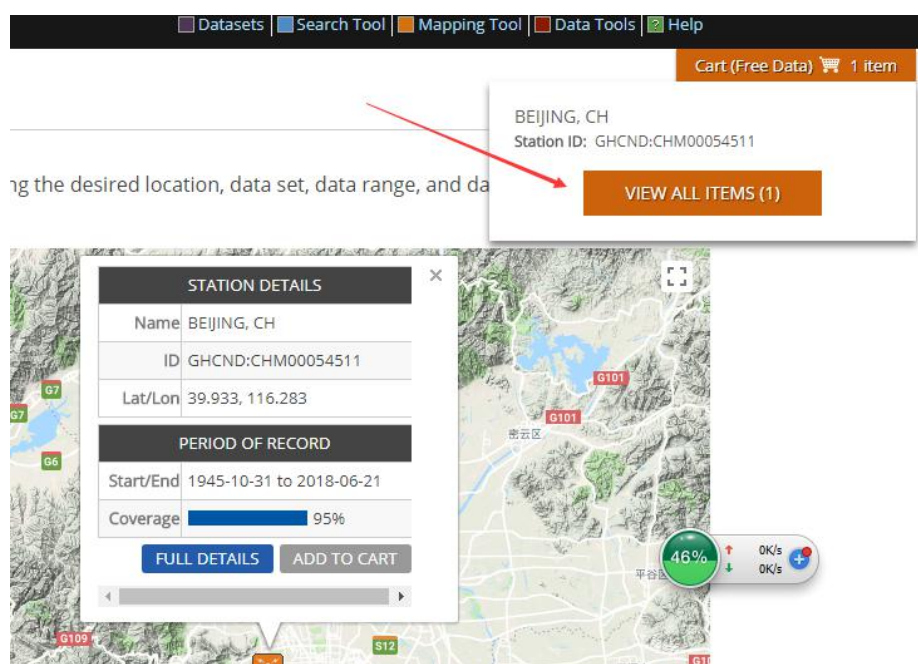
选择数据要素

接下来, 点右侧地图上的图标



点击“ADD TO CART”后，在购物车里就会有条订单数据

点右上角的购物车



选择 VIEW ALL ITEMS(1)， 出现如下画面

Custom GHCN-Daily CSV
(Additional options available on next page)

Custom GHCN-Daily Text
(Additional options available on next page)

选择csv格式

Select the Date Range
Click to choose the date range below.

2000-06-01 to 2018-06-01

重新确认一下日期区间

Review the items in your cart

[CLEAR CART]

BEIJING, CH
View Full Details
Station ID: GHCND:CHM00054511
Period of Record: 1945-10-31 : 2018-06-21

Delete

CONTINUE

点“CONTINUE”后，出现更多数据选项

首先是站点的属性，可以选择数据是否包括站点名称，坐标，单位格式等。

Step 1: Choose Options Step 2: Review Order Step 3: Order Complete

Data types are grouped by category for easier selection and can be selected as a group or individually. Selected data types will be included in the customized output.

Station Detail & Data Flag Options

Additional output options such as data flags (attributes), station names, and geographic location are also available.

☒ Station Name
☒ Geographic Location
☒ Include Data Flags

Units: Standard
Standard
Metric

Select data types for custom output

具体的数据格式选项如下所示：

Select data types for custom output

The items below are data types that can be added to the output. Expand the data type category headers to view the categorized data type names and descriptions.

[Show All / Hide All](#) | [Select All / Deselect All](#)

☒ ☐ Precipitation

☒ Precipitation (PRCP)
 ☒ Snow depth (SNWD)

☒ ☐ Air Temperature

☒ Average Temperature. (TAVG)
 ☒ Maximum temperature (TMAX)
 ☒ Minimum temperature (TMIN)

BACK
CONTINUE

因为在前面的界面中，只选择了降水和空气温度两个要素，所以现在出现的是对这两个要素的详细信息，勾选后，点“CONTINUE”
下面的画面就是订单预览界面：

REQUESTED DATA REVIEW	
Dataset	Daily Summaries → 数据集= 日报数据
Order Start Date	2000-06-01 00:00 →
Order End Date	2018-06-01 23:59 → 数据范围
Output Format	Custom GHCN-Daily CSV
Data Types	PRCP, SNWD, TAVG, TMAX, TMIN → 数据要素
Custom Flag(s)	Station Name, Geographic Location, Include Data Flags
Units	Standard
Stations/Locations	BEIJING, CH (Station ID: GHCND:CHM00054511)

Enter email address

Please enter your email address. This is the address to which your data links and information regarding this order will be sent. Please read [NOAA's Privacy Policy](#) if you have any concerns.

Email Address
→ 输入接收数据的邮箱

goodbeam@163.com

Enter email address

Please enter your email address. This is the address to which your data links and information regarding this order will be sent. Please read [NOAA's Privacy Policy](#) if you have any concerns.

Email Address

Verify Email Address

☒ Remember my email address

[Uncheck to forget]

NOAA will not share your email address with anyone. The email address will not be used for any purpose other than communicating the order status.

[EDIT ORDER](#)[SUBMIT ORDER](#)


就是输入自己的邮箱地址, 点 SUBMIT ORDER

Request Submitted

Step 1: Choose Options

Step 2: Review Order

Step 3: Order Complete



Your request was successfully submitted.
An email with a link to the requested data should be sent shortly.

[Print Receipt](#)

ORDER INFORMATION	
Order Number	1380937
Order Format	Custom GHCN-Daily CSV
Email Address	goodbeam@163.com
Date Submitted	2018-06-24 06:51 EDT
Check Order Status	CHECK ORDER STATUS

PERIOD OF REQUEST

过会就可以在邮箱中收取数据了.

任务 7. 安装数据分析相关的依赖包

任务描述

内 容： 安装本项目中用到的 python 包

学 时： 3 学时

知识点： pandas, numpy, matplotlib 和 Statsmodels 包的介绍

重点： 如何通过 pip 命令安装倚赖包

难点：

任务指导

- 1, 通过 pip 命令安装 python 包
- 2, 依赖包的功能介绍

任务实现

1、Pip 命令

pip 是一个 Python 包管理工具,主要是用于安装 PyPI 上的软件,pip 是 python3 的自带命令,不需要额外安装。

打开命令行窗口,输入 `pip install 包名` ,就可以了

比如,要安装 pandas 包,直接输入 `pip install pandas`,系统会自动联网下载 pandas 包及相关的文件。

2、依赖包功能介绍

➤ Statsmodels

Statsmodels 是 Python 的统计建模和计量经济学的工具包,包括一些描述统计、统计模型估计和推断。包括线性回归模型、一般线型模型、robust linear models、离散选择模型(logit)、ANOVA、时间序列分析、非参检验、各种统计检验、读取各种格式的数据及输出、绘图等。

➤ Pandas

Pandas 是基于 NumPy 的一种工具,该工具是为了解决数据分析任务而创建的。Pandas 纳入了大量第三方库和一些标准的数据模型,提供了高效地操作大型数据集所需的工具。Pandas 是最具有统计意味的工具包,某些方面优于 R 软件。数据结构有一维的 Series,二维的 DataFrame(类似于 Excel 或者 SQL 中的表)。

➤ Numpy

NumPy 是 Python 用于处理大型矩阵的一个速度极快的数学库。它允许你在 Python 中做向量和矩阵的运算,而且很多底层的函数都是用 C 写的,你将获得在普通 Python 中无法达到的运行速度。这是由于矩阵中每个元素的数据类型都是一样的,这也就减少了运算过程中的类型检测。

Numpy 是 Python 的一个科学计算的库,提供了矩阵运算的功能,其一般与 Scipy、Matplotlib 一起使用。

➤ Matplotlib

Python 中最著名的绘图系统。matplotlib 的复杂性给其带来了很强的定制功能。一般需要掌握的绘图模型包括:散点图,折线图,条形图,直方图,饼状图,箱形图的绘制等。

任务 8. 虚拟机配置

任务描述

内 容: 配置 ubuntu 虚拟机,方便后续的 spark 集群配置

学 时: 3 学时

知识点: 配置 ubuntu 虚拟机

重点:

难点:

任务指导

任务实现

1、安装虚拟机

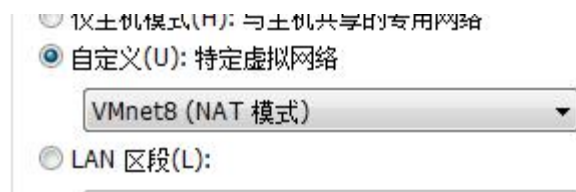
首先需要在 windows 上安装 vmware 和 ubuntu 虚拟机,

vmware 下载地址: <https://www.vmware.com/>

ubuntu 下载地址: <http://cdimage.ubuntu.com/daily-live/current/>

2、配置虚拟机的网络

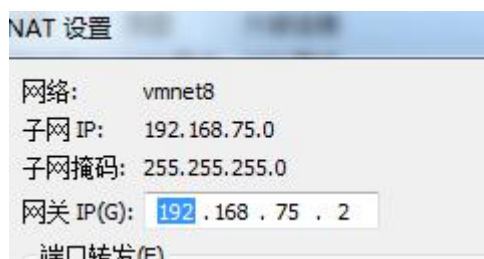
首先, 将虚拟机的网络设置设置为自定义, 选择 VMnet8:



随后, 我们点击 VMWARE 上的编辑-虚拟网络编辑-右下角的更改设置, 应该有三个连接方式, 这里我们把其他两个移除, 只剩下 VMnet8:

名称	类型	外部连接	主机连接	DHCP	子网地址
VMnet8	NAT 模式	NAT 模式	已连接	-	192.168.75.0

随后, 点击 NAT 设置, 我们可以发现网关是 192.168.75.2



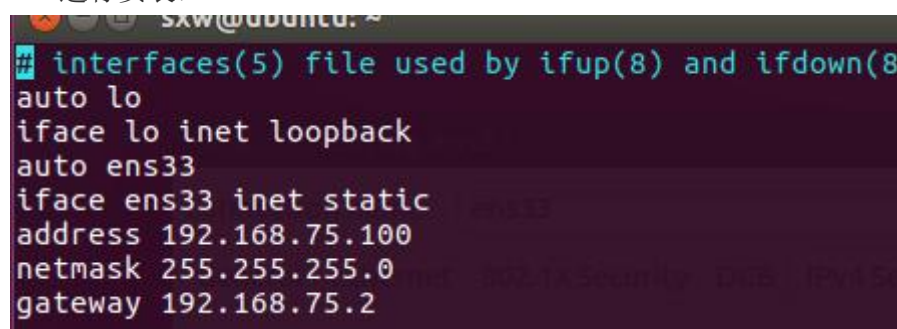
接下来, 我们要设置虚拟机的 ip: 点击右上角的 edit connectinos, 设置 Ipv4, 如下图所示:



随后修改两个文件：

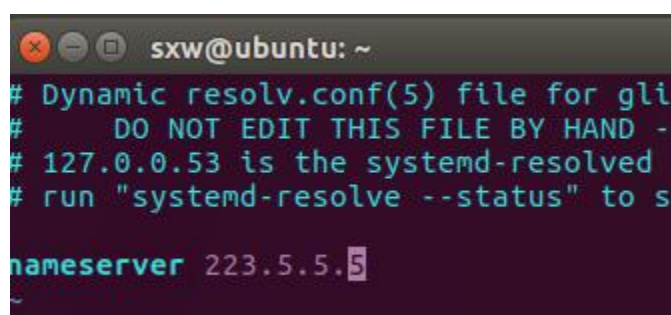
修改 interfaces 文件

命令：sudo vim /etc/network/interfaces (如果没有 vim 命令，使用 sudo apt-get install vim 进行安装)：



修改 resolv.conf 文件

命令：sudo vim /etc/resolv.conf:



接下来重启我们的网络就可以啦

命令：sudo /etc/init.d/networking restart

(如果启动失败，重启虚拟机即可)

查看我们的 ip，使用 ifconfig 命令，如果没有安装(使用 sudo apt install net-tools 进行安装):


```

sxw@ubuntu:~$ ifconfig
ens33: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.75.100 netmask 255.255.255.0 broadcast 192.168
    inet6 fe80::20c:29ff:fec6:b1c4 prefixlen 64 scopeid 0x20<li
    ether 00:0c:29:c6:b1:c4 txqueuelen 1000 (Ethernet)
    RX packets 9873 bytes 9033028 (9.0 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 4380 bytes 348750 (348.7 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

```

3、使用 xshell 连接本地虚拟机

下载 xshell，百度搜索 xshell，使用百度提供的下载地址即可。

要想使用 xshell 的 ssh 方式访问虚拟机，首先要在虚拟机上安装 ssh 服务

使用命令：

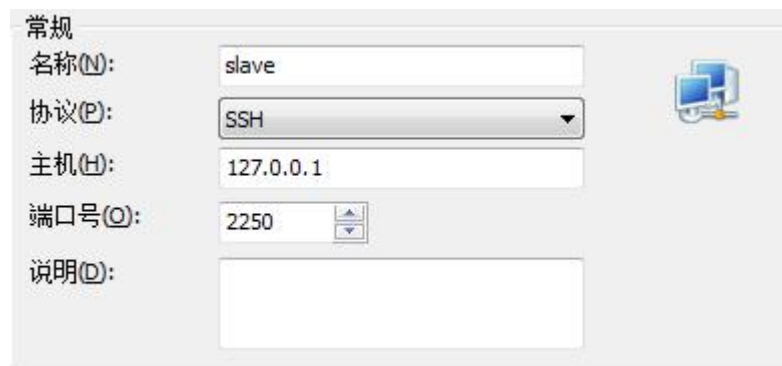
`sudo apt-get install openssh-server`

随后启动 ssh 服务：`sudo /etc/init.d ssh start`

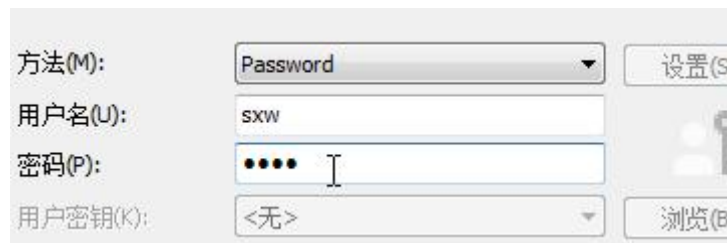
再次点击 VMWARE 上的编辑-虚拟网络编辑-右下角的更改设置，设置端口转发：



随后打开 xshell，新建连接：



设置用户名和密码：



随后点击连接即可，发现连接成功！

```

Last login: Mon Jun  5 20:04:32 2017 from 192.168.75.2
sxw@ubuntu:~$ █

```

4、修改虚拟机主机名以及 hosts 文件

这里以修改主节点主机名称为例，其他节点类似。

使用命令：`sudo vim /etc/hostname` 查看当前主机名，并修改为 `master`：
重启之后生效：

```
sxw@master:/etc$ sudo vim /etc/hostname
sxw@master:/etc$
```

两个从节点的主机依次修改为 `slave1`，`slave2`

接下来，将主节点和两个从节点的 `ip` 和主机名添加到 `hosts` 文件中
使用命令

`sudo vim /etc/hosts`

修改的结果为：

```
terminal File Edit View Search Terminal
sxw@master:/etc
127.0.0.1    localhost
192.168.75.50 master
192.168.75.75 slave1
192.168.75.100 slave2
```

两个从节点的 `hosts` 文件修改为同样的结果，此时发现各虚拟机之间可以 `ping` 通

```
sxw@master:/etc$ ping slave1
PING slave1 (192.168.75.75) 56(84) bytes of data.
64 bytes from slave1 (192.168.75.75): icmp_seq=1 ttl=64 time=0.315 ms
64 bytes from slave1 (192.168.75.75): icmp_seq=2 ttl=64 time=0.181 ms
^C
```

5、配置 SSH 免验证登录

接下来，需要让主节点可以免验证的登录到从节点，从而在进行任务调度时可以畅通无阻。

首先要在各个节点上生成公钥和私钥文件，这里以 `slave1` 节点进行讲解，其他节点操作方式完全相同。

我们首先要开启 `ssh` 服务，使用命令：`sudo /etc/init.d/ssh start`

```
sxw@slave1:~$ sudo /etc/init.d/ssh start
[ ok ] Starting ssh (via systemctl): ssh.service.
```

随后使用如下命令生成公钥和私钥文件：

`ssh-keygen -t rsa -P ""`

```

sxw@slave1:~$ ssh-keygen -t rsa -P ""
Generating public/private rsa key pair.
Enter file in which to save the key (/home/sxw/.ssh/id_rsa):
Created directory '/home/sxw/.ssh'.
Your identification has been saved in /home/sxw/.ssh/id_rsa.
Your public key has been saved in /home/sxw/.ssh/id_rsa.pub.
The key fingerprint is:
SHA256:70W6xyrTQl030Xt3bqjZ/t/jTBoA+KdtEWYo0fHnetw sxw@slave1
The key's randomart image is:
+---[RSA 2048]---+
|                |
|      . . . .   |
|      o.o       |
|    o o = o     |
|      o = * .   |
|    So =.+ .   |
|    o.+o* o +   |
|      . +o=.+ Eo|
|    +.oo+o* .+  |
|      ++oo++=*  |
+-----[SHA256]-----+

```

在所有节点上生成密钥文件之后，我们需要将从节点的公钥传输给主节点，使用命令：

```
cd ~/.ssh
```

```
scp id_rsa.pub sxw@master:~/.ssh/id_rsa.pub.slave1
```

```

sxw@slave1:~$ cd ~/.ssh
sxw@slave1:~/.ssh$ scp id_rsa.pub sxw@master:~/.ssh/id_rsa.pub.slave1
The authenticity of host 'master (192.168.75.50)' can't be established.
ECDSA key fingerprint is SHA256:UXEvGTsH3HQvi0p/0jphQYKib4Z0EZTq7lbBKlk2FKU.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'master,192.168.75.50' (ECDSA) to the list of known hosts.
sxw@master's password:
id_rsa.pub                                100% 392      1.7MB/s   00:00

```

随后，在主节点下，将所有的公钥信息拷贝到 `authorized_keys` 文件下：

使用命令：

```
cd ~/.ssh
```

```
cat id_rsa.pub >> authorized_keys
```

```
cat id_rsa.pub.slave1 >> authorized_keys
```

```
cat id_rsa.pub.slave2 >> authorized_keys
```

```

sxw@master:/etc$ cd ~/.ssh
sxw@master:~/.ssh$ ls
authorized_keys  id_rsa.pub          id_rsa.pub.slave2
id_rsa          id_rsa.pub.slave1  known_hosts
sxw@master:~/.ssh$ cat id_rsa.pub >> authorized_keys
sxw@master:~/.ssh$ cat id_rsa.pub.slave1 >> authorized_keys
sxw@master:~/.ssh$ cat id_rsa.pub.slave2 >> authorized_keys

```

接下来将 `authorized_keys` 文件复制到 `slave1` 和 `slave2` 节点目录下：

```
scp authorized_keys sxw@slave1:~/.ssh/authorized_keys
```



```
scp authorized_keys sxw@slave2:~/ssh/authorized_keys
```

```
sxw@master:~/.ssh$ scp authorized_keys sxw@slave1:~/.ssh/authorized_keys
The authenticity of host 'slave1 (192.168.75.75)' can't be established.
ECDSA key fingerprint is SHA256:30lXVHnNmFowkW14ojR8V8z+EZj+lIt/1dS1Qt9vh/Y.
Are you sure you want to continue connecting (yes/no)? yes
Warning: Permanently added 'slave1,192.168.75.75' (ECDSA) to the list of known h
osts.
sxw@slave1's password:
authorized keys                               100% 2352      7.4MB/s   00:00
```

接下来我们验证是否可以免密码登录：使用命令
ssh slave1

```
sxw@master:~/.ssh$ ssh slave1
Welcome to Ubuntu Artful Aardvark (development branch) (GNU/Linux 4.10.0-21-gene
ric x86_64)

* Documentation:  https://help.ubuntu.com
* Management:    https://landscape.canonical.com
* Support:       https://ubuntu.com/advantage

* Ubuntu 12.04 LTS ('precise') end-of-life was April 28, 2017
  ongoing security updates for 12.04 are available with Ubuntu Advantage
  - https://ubu.one/U1204esm
* Aaron Honeycutt from the Kubuntu Council on art and design in Kubuntu
  - https://ubu.one/kubuart
* The Ubuntu Desktop team wants your feedback on the move to Gnome
  - https://ubu.one/2GNome

81 packages can be updated.
0 updates are security updates.

sxw@slave1:~$
```







6、安装 JDK1.8

这里我们可以直接使用 linux 的命令下载 jdk，当然也可以在本地下载之后传输到虚拟机中，到 java 官网中下载最新的 jdk 文件即可。

刚才我们配置了端口转发，因此我们可以使用 winscp 进行文件传输：



传输文件到/home/sxw/Documents 路径下，直接将文件进行拖拽即可：

/home/sxw/Documents				
名字	大小	已改变	权限	拥有者
		2017/6/6 14:04:40	rwxr-xr-x	sxw
 hadoop		2016/8/18 9:49:57	rwxr-xr-x	sxw
 hdfs		2017/6/6 13:09:51	rwxr-xr-x	sxw
 jdk		2017/3/15 16:35:35	rwxr-xr-x	sxw
 hadoop-2.7.3.tar.gz	209,075...	2017/6/2 13:22:54	rw-rw-r--	sxw
 jdk-8u131-linux-x64...	181,192...	2017/6/2 12:05:01	rw-rw-r--	sxw

随后，在该路径下，使用如下命令进行解压：

tar -zxvf 文件名

```
jdk1.8.0_131/README.html
sxw@slave1:~/Documents$ tar -zxvf jdk-8u131-linux-x64.tar.gz
```

重命名 jdk 文件夹为 jdk

```
jdk1.8.0_131/
sxw@slave1:~/Documents$ mv jdk1.8.0_131 jdk
```

随后修改配置文件：

sudo vim /etc/profile

添加如下三行：

export JAVA_HOME=/home/sxw/Documents/jdk

export CLASSPATH=.:\$JAVA_HOME/lib:\$JAVA_HOME/jre/lib:\$CLASSPATH

export PATH=\$JAVA_HOME/bin:\$JAVA_HOME/jre/bin:\$PATH

```
sxw@slave1: ~/Documents
# /etc/profile: system-wide .profile file for the Bourne shell (sh(1)
# and Bourne compatible shells (bash(1), ksh(1), ash(1), ...).

export JAVA_HOME=/home/sxw/Documents/jdk
export CLASSPATH=.:$JAVA_HOME/lib:$JAVA_HOME/jre/lib:$CLASSPATH
export PATH=$JAVA_HOME/bin:$JAVA_HOME/jre/bin:$CLASSPATH
```

使用 source 命令使修改生效，同时查看是否安装成功

```
sxw@slave1:~/Documents/jdk$ source /etc/profile
sxw@slave1:~/Documents/jdk$ java -version
java version "1.8.0_131"
Java(TM) SE Runtime Environment (build 1.8.0_131-b11)
Java HotSpot(TM) 64-Bit Server VM (build 25.131-b11, mixed mode)
sxw@slave1:~/Documents/jdk$
```

任务 9. 配置 Spark 集群

任务描述

内 容： 配置一个 spark 集群系统

学 时： 3 学时

知识点： spark 集群系统

重点：

难点：

初步了解 spark 集群的概念

任务指导

任务实现

1、安装 hadoop

我们首先在主节点上配置好 hadoop 的文件，随后使用 scp 命令传输到从节点上即可
同样，我们在 hadoop 官网下载 hadoop 文件，通过 winscp 传入主节点，使用 tar 命令进行解压，并修改文件夹名为 hadoop，这里暂且略过。

修改环境变量（所有节点都需要修改）并使用 source 命令使其生效：

```
export HADOOP_HOME=/home/sxw/Documents/hadoop
export PATH=$HADOOP_HOME/bin:$HADOOP_HOME/sbin:$PATH
export HADOOP_CONF_DIR=$HADOOP_HOME/etc/hadoop
export YARN_CONF_DIR=$HADOOP_HOME/etc/hadoop
```

接下来修改 hadoop 的配置文件：

打开 \$HADOOP_HOME/etc/hadoop/hadoop-env.sh

修改 JAVA_HOME 如下：

```
export JAVA_HOME=/home/sxw/Documents/jdk
```

打开 \$HADOOP_HOME/etc/hadoop/slaves

修改 slaves 里添加两个从节点的名称

slave1

slave2

打开\$HADOOP_HOME/etc/hadoop/core-site.xml

```
<configuration>
  <property>
    <name>fs.defaultFS</name>
    <value>hdfs://master:9000</value>
  </property>
  <property>
    <name>io.file.buffer.size</name>
    <value>131072</value>
  </property>
  <property>
    <name>dfs.permissions</name>
    <value>false</value>
  </property>
</configuration>
```

打开\$HADOOP_HOME/etc/hadoop/hdfs-site.xml

```
<configuration>
  <property>
    <name>dfs.namenode.secondary.http-address</name>
    <value>master:50090</value>
  </property>
  <property>
    <name>dfs.replication</name>
    <value>2</value>
  </property>
  <property>
    <name>dfs.namenode.name.dir</name>
    <value>file:/home/sxw/Documents/hdfs/tmp/name</value>
  </property>
  <property>
    <name>dfs.datanode.data.dir</name>
    <value>file:/home/sxw/Documents/hdfs/tmp/data</value>
  </property>
  <property>
    <name>dfs.permissions</name>
    <value>false</value>
  </property>
</configuration>
```

打开\$HADOOP_HOME/etc/hadoop/mapred-site.xml

首先使用如下命令生成 mapred-site.xml 文件:

```
cp mapred-site.xml.template mapred-site.xml
```

随后进行修改:


```

<configuration>
  <property>
    <name>mapred.job.tracker</name>
    <value>master:9001</value>
  </property>
  <property>
    <name>mapreduce.framework.name</name>
    <value>yarn</value>
  </property>
  <property>
    <name>mapreduce.jobhistory.address</name>
    <value>master:10020</value>
  </property>
  <property>
    <name>mapreduce.jobhistory.address</name>
    <value>master:19888</value>
  </property>
</configuration>

```

打开\$HADOOP_HOME/etc/hadoop/yarn-site.xml

```

<configuration>
<!-- Site specific YARN configuration properties -->
  <property>
    <name>yarn.nodemanager.aux-services</name>
    <value>mapreduce_shuffle</value>
  </property>
  <property>
    <name>yarn.resourcemanager.address</name>
    <value>master:8032</value>
  </property>
  <property>
    <name>yarn.resourcemanager.scheduler.address</name>
    <value>master:8030</value>
  </property>
  <property>
    <name>yarn.resourcemanager.resource-tracker.address</name>
    <value>master:8031</value>
  </property>
  <property>
    <name>yarn.resourcemanager.admin.address</name>
    <value>master:8033</value>
  </property>
  <property>
    <name>yarn.resourcemanager.webapp.address</name>
    <value>master:8088</value>
  </property>
</configuration>

```

至此，hadoop 的配置文件就修改完了，我们用 scp 命令将修改好的 hadoop 文件传入到子节点即可

2、安装 spark

我们首先在主节点上配置好 spark 的文件，随后使用 scp 命令传输到从节点上即可。同样在 spark 官网下载最新的 spark 文件，并使用 winscp 传入虚拟机，使用 tar 命令进行解压，并重命名文件夹为 spark。
添加 spark 到环境变量并使其生效：

```

export SPARK_HOME=/home/sxw/Documents/spark
export PATH=$SPARK_HOME/bin:$PATH

```


打开\$SPARK_HOME/conf/spark-env.sh
 首先使用如下命令生成 spark-env.sh 文件：
 cp spark-env.sh.template spark-env.sh
 随后进行修改：

```
export SCALA_HOME=/home/sxw/Documents/scala
export JAVA_HOME=/home/sxw/Documents/jdk
export SPARK_MASTER_IP=master
export SPARK_WORKER_MEMORY=1g
export HADOOP_CONF_DIR=/home/sxw/Documents/hadoop/etc/hadoop
```

打开\$SPARK_HOME/conf/slaves
 首先使用如下命令生成 slaves 文件：
 cp slaves.template slaves
 随后进行修改：

```
# A spark worker will be started on each of the machines listed below.
master
slave1
slave2
```

至此，spark 的配置文件就修改完了，我们用 scp 命令将修改好的 spark 文件传入到子节点即可，不要忘记修改子节点的环境变量。

3、集群启动和关闭

首先我们编写一个启动脚本

```
#!/bin/bash
echo -e "\033[31m =====Start The Cluster===== \033[0m"
echo -e "\033[31m Starting Hadoop Now !!! \033[0m"
/home/sxw/Documents/hadoop/sbin/start-dfs.sh
/home/sxw/Documents/hadoop/sbin/start-yarn.sh
echo -e "\033[31m Starting Spark Now !!! \033[0m"
/home/sxw/Documents/spark/sbin/start-all.sh
echo -e "\033[31m The Result of The Command \"jps\" : \033[0m"
jps
echo -e "\033[31m =====END===== \033[0m"
```

可以看到，hadoop 的启动需要两个命令，分别启动 dfs 和 yarn，传统的 start-all.sh 已经被弃用。而 spark 的启动只需要一个命令。

启动的结果如下图所示：

```

Documents/spark/logs/spark-sxw-org.apache.
The Result Of The Command "jps" :
7376 NameNode
8144 Jps
7841 Master
7603 SecondaryNameNode
7753 ResourceManager
8111 Worker
=====END=====

```

随后我们再编写一个关闭集群的脚本：

```

#!/bin/bash
echo -e "\033[31m ===== stoping The cluster ===== \033[0m"
echo -e "\033[31m stoping Spark Now !!! \033[0m"
/home/sxw/Documents/spark/sbin/stop-all.sh
echo -e "\033[31m stoping Hadoop Now !!! \033[0m"
/home/sxw/Documents/hadoop/sbin/stop-dfs.sh
/home/sxw/Documents/hadoop/sbin/stop-yarn.sh
echo -e "\033[31m The Result of The Command \"jps\" : \033[0m"
jps
echo -e "\033[31m =====END===== \033[0m"

```

4、Hadoop 测试

我们在/home/sxw/Documents 下建立一个 wordcount.txt 文件
文件内容如下图：

```

Hello hadoop
hello spark
hello bigdata

```

到 hadoop 的 bin 路径下执行如下三条命令：

```
hadoop fs -mkdir -p /Hadoop/Input
```

```
hadoop fs-put wordcount.txt /Hadoop/Input
```

```
hadoop jar
```

```
/home/sxw/Documents/hadoop/share/hadoop/mapreduce/hadoop-mapreduce-examples-2.7.3.jar wordcount /Hadoop/Input /Hadoop/Output
```

可以看到我们的 hadoop 在进行运算了

```

sxw@master:~/Documents$ cd hadoop/bin
sxw@master:~/Documents/hadoop/bin$ hadoop fs -mkdir -p /Hadoop/Input
sxw@master:~/Documents/hadoop/bin$ hadoop fs -put /home/sxw/Documents/wordcount.
txt /Hadoop/Input
sxw@master:~/Documents/hadoop/bin$ hadoop jar /home/sxw/Documents/hadoop/share/h
adoop/mapreduce/hadoop-mapreduce-examples-2.7.3.jar wordcount /Hadoop/Input /Had
oop/Output

```

我们直接利用 `spark-shell` 进行测试，编写几条简单 `pyspark` 语句：
到 `spark` 的 `bin` 路径下执行 `./pyspark` 命令进入 `pyspark` 的交互模式，

[illegible]

任务 10. 在 spark 中进行数据清洗

内 容: 使用 spark 进行数据清洗
学 时: 3 学时
知识点: spark 脚本的使用
重点:
难点:

初步了解 spark 中数据处理的概念, spark shell 支持 scala 或 python, 在这个教学任务中, 我们使用 python 脚本来执行相关的 shell 命令

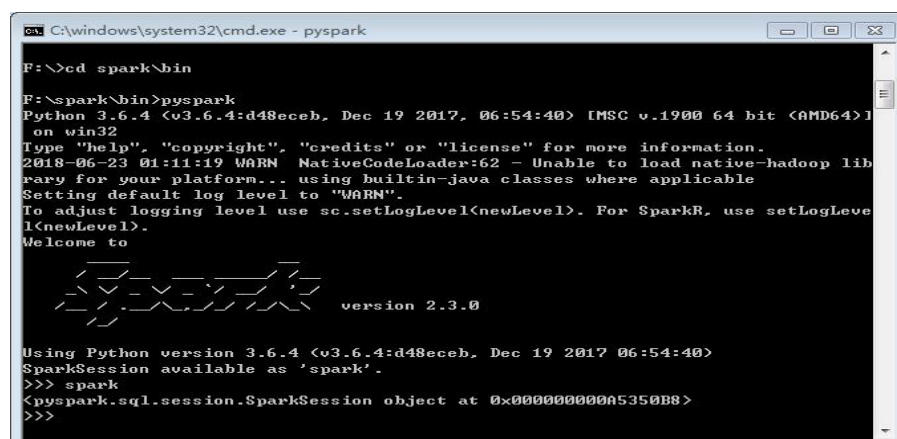
任务指导

- 1、加载原始数据
- 2、进行数据处理

任务实现

- 1、加载原始数据

进入 F:\spark\bin 目录, 运行 pyspark, 进入 spark



导入依赖包

```
import pandas as pd
from datetime import datetime    #系统自带
from dateutil import parser
```

加载历史数据文件

```
data_raw = pd.read_csv('00054511.csv', encoding='utf-8') #默认路径是 bin 目录, 或者 f:\\00054511.csv
```

原始的数据格式如下所示:

A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q
STATION	NAME	LATITUDE	LONGITUDE	ELEVATION	DATE	PRCP	PRCP_ATT	SNWD	SNWD_ATT	TAVG	TAVG_ATT	TMAX	TMAX_ATT	TMIN	TMIN_ATT	ATTRIBUTES
CHM00054	BEIJING,	39.933	116.283	55	1980-1-1	0,,s				22,,s		29,,s		15,,s		
CHM00054	BEIJING,	39.933	116.283	55	1980-1-2	0.02,,s				27,,s		33,,s		22,,s		
CHM00054	BEIJING,	39.933	116.283	55	1980-1-3	0,,s				29,,s		33,,s		25,,s		
CHM00054	BEIJING,	39.933	116.283	55	1980-1-4	0,,s				22,,s		31,,s		16,,s		
CHM00054	BEIJING,	39.933	116.283	55	1980-1-5	0,,s				20,,s		35,,s		7,,s		
CHM00054	BEIJING,	39.933	116.283	55	1980-1-6	0,,s				24,,s		31,,s		15,,s		
CHM00054	BEIJING,	39.933	116.283	55	1980-1-7	0,,s				21,,s		37,,s		7,,s		
CHM00054	BEIJING,	39.933	116.283	55	1980-1-8	0,,s				24,,s		35,,s		14,,s		
CHM00054	BEIJING,	39.933	116.283	55	1980-1-9	0,,s				26,,s		31,,s		18,,s		
CHM00054	BEIJING,	39.933	116.283	55	1980-1-10	0,,s				18,,s		26,,s		13,,s		
CHM00054	BEIJING,	39.933	116.283	55	1980-1-11	0,,s				17,,s		31,,s		5,,s		
CHM00054	BEIJING,	39.933	116.283	55	1980-1-12	0,,s				28,,s		39,,s		15,,s		
CHM00054	BEIJING,	39.933	116.283	55	1980-1-13	0,,s				29,,s		38,,s		18,,s		

2、筛选数据

配置数据属性

```
data_raw['date'] = data_raw['DATE'].apply(parser.parse)      #把 DATE 列解析成日期格式
data_raw['tmax'] = data_raw['TMAX'].astype(float)
data_raw['tmin'] = data_raw['TMIN'].astype(float)
```

得到需要的列，过滤掉空值

```
data = data_raw.loc[:, ['date', 'tmax', 'tmin']]             #筛选出需要的三个列
data = data[(pd.Series.notnull(data['tmax'])) & (pd.Series.notnull(data['tmin']))]    #把
空数据过滤掉
```

根据过滤条件，进一步筛选出符合条件的数据

```
data = data[(data['date'] >= datetime(1980,1, 1)) & (data['date'] <= datetime(2016,1, 1))]
data.query("date.dt.day == 28 & date.dt.month == 6", inplace=True) #得到历史上每年 6.28 的数据
```

把最终结果写入 csv 文件

```
data.to_csv('maxmin.csv', index=None)
```

任务 11. ARIMA 模型研究

任务描述

内 容： 介绍关于时间序列的相关算法和模型

学 时： 3 学时

知识点： 时间序列知识, ARMA 和 ARIMA 模型

重点：

难点：

了解 AR、MA、ARMA、ARIMA 模型的原理和实现思路

任务指导

- 1, ARIMA 模型简介
- 2, AR 概念
- 3, MA 概念

任务实现

任务 12. 建立基于时间序列的数据模型

任务描述

内 容： 基于时间序列算法, 利用 python 的数据分析包, 建立数据模型

学 时： 3 学时

知识点： 时间序列知识, ARIMA 模型

重点：

难点： 利用差分得到平稳数据

pandas 和 statsmodels 都提供了时间序列相关分析功能, 本示例使用 pandas 进行数据整理, 使用 statsmodels 做时间序列预测应用。有关时间序列算法的选择, 实际场景中最常用的是 ARIMA 或 ARMA 了, 因此本示例将使用 ARIMA/ARMA 来做时间序列分析, ARIMA 相对于 ARMA 多了一个差分的过程

对于这两种时间序列方法而言, 应用的难点是如何根据不同的场景判断参数值 (即 p 、 d 、 q)。讲师需要带领学生完成一种通用模型以供参考, 其余的可以自己完成。

任务指导

- 1, 导入需要的 python 包
- 2, 准备训练数据
- 3, 展示原始数据
- 4, 根据实际情况进行差分
- 5, 选择合适的 ARIMA 模型
- 6, 验证模型
- 7, 生成预测数据

任务实现

任务 13. 创建 python 处理类

任务描述

内 容: 利用数据模型, 创建 python 接口程序

学 时: 6 学时

知识点: python 读写文件操作, 生成 json 格式数据

重点:

难点:

新建 python 类, 做成标准化的处理程序供其它模块调用

任务指导

- 1, 定义处理类
- 2, 设计处理函数
- 3, 调用方式

任务实现

任务 14. Json 数据格式处理

任务描述

内 容：对预测数据进行处理，生成标准化的 json 数据格式

学 时：2 学时

知识点：python 处理 json 程序，python 中日期函数的使用

重点：

难点：

任务指导

1, Json 处理

任务实现

任务 15. 数据传输

任务描述

内 容：传输预测图形和 json 格式的预测数据

学 时：2 学时

知识点：python 中的网络数据传输

重点:

难点:

教授给学生如何把预测数据传输出去

任务指导

1, socket 数据传输

任务实现

任务 16. 配置 Flask 运行环境

任务描述

内 容: 介绍 Flask 框架的全局配置; 使用 pip 安装

学 时: 2 学时

知识点: python 配置, flask 架构

重 点: 初步了解 Flask 架构

难 点: Flask 框架的运行原理

讲授学生 python 框架 Flask 的基本原理和运行方式, 利用 Flask 创建一个微服务接口, 通过传递一些必要的参数, 达到数据自动处理的目的

任务指导

1. Flask 概述及学习目的和意义
2. Pip 安装 Flask 及其依赖包
3. 启动 Flask 程序

任务实现

任务 17. 配置 Flask 路由

任务描述

内 容：介绍 Flask 框架中路由的知识
学 时： 2 学时
知识点： python 中路由的配置
重 点：
难 点：

讲授学生 Flask 框架中的路由的概念，包括基本原理和运行方式
在 Flask 框架中，路由表示用户请求的 URL 找出其对应的处理函数

任务指导

1. 注册路由
2. 为路由指定支持的 http 方法(get 或者 post)
3. 动态匹配 URL

任务实现

任务 18. 微服务架构设计

任务描述

内 容：利用 Flask 框架，搭建一个微服务数据处理接口

学 时：2 学时

知识点：Flask 框架, 使用路由技术

重点：

难点：

教授给学生如何通过参数传递的方式, 调用 python 接口程序, 把生成的预测数据传输出去

任务指导

- 1, 定义路由接口
- 2, 定义要传递的参数
- 3, 定义数据处理方法

任务实现

任务 19. Java IDE 配置

任务描述

内 容：MyEclipse 环境配置, tomcat 部署知识

学 时：2 学时

知识点：

重点：

难点：

任务指导

- 1, 下载 myeclipse
- 2, 安装配置 myeclipse
- 3, 配置 tomcat

任务实现

任务 20. Java 程序调用微服务接口

任务描述

内 容: 利用 java 调用 Python 的微服务框架

学 时: 2 学时

知识点: java 中的 Http 请求

重点:

难点:

教授给学生如何通过 java 程序, 通过传递一些必要的参属调用 python 接口程序, 得到最新的预测数据

任务指导

- 1, 定义 java 类
- 2, 定义要传递的参数
- 3, 定义 java 接收数据的方法

任务实现

任务 21. 用户登录

任务描述

内 容：用户登录设计

学 时：4 学时

知识点：spring MVC 知识，前台页面和后台数据的交互

重点：Spring 前后台交互的原理

难点：Hibernate 知识

这个是 java 端的起始程序, 可以从前台页面开始逐行代码讲解, 以后有类似的操作可以交给学生自己完成, 初步了解下 springmvc 的工作原理, 通过实际的例子来了解。

任务指导

- 1, 登录页面的实现
- 2, 登录验证码设计
- 3, springmvc 的页面交互过程
- 4, easyui 的初步认识

任务实现

任务 22. 数据展示功能

任务描述

内 容： 展示预测分析的图片 and 列表数据

学 时： 4 学时

知识点： websocket 知识, json 数据展示

重点：

难点：

任务指导

- 1、Websocket 知识介绍
- 2、Websocket 方法和事件
- 3、Html 网页布局

任务实现

任务 23. 形成一周天气预测数据

任务描述

内 容： 多次调用微服务接口, 形成未来七天的天气数据, 并形成走势图

学 时： 4 学时

知识点： Echart 图表知识, json 数据展示

重点：

难点:

任务指导

任务实现

任务 24. 用户管理功能

任务描述

内 容: 系统用户的增, 删, 改, 查功能

学 时: 3 学时

知识点: EasyUI 对数据的管理功能

重点: 理解用户管理过程中整个数据前后交互以及如何用 easyui 来实现

难点: easyui 的各种标签以及验证, 图片上传

任务指导

- 1、用户增删改查的整个前后台的交互问题
- 2、Easyui 在用户管理系统的应用。

3、图片上传的过程。

任务实现

任务 25. 部门管理

任务描述

内 容：部门信息的增，删，改，查功能

学 时：3 学时

知识点：spring, springmvc, hibernate 之间数据的交互，easyui 的应用

重点： spring, springmvc, hibernate 的用法，hql 和 sql 的注意问题

难点： easyui 的熟悉

任务指导

- 1、部门数据展示
- 2、部门数据的修改

任务实现

任务 26. 角色管理功能

任务描述

内 容：完成角色的增删改查功能

学 时：3 学时

知识点：数据列表展示，权限点的设置，删除角色时的注意事项

重点：角色和权限点的对应关系

难点：权限点的设置，easyui 的树形结构的展示，以及 treenode 的封装问题

任务指导

1. 新建角色列表页面
2. 完成后台 java 类的设计
3. 完成前端 easyui 的实现

任务实现

任务 27. 权限管理功能

任务描述

内 容：完成角色的增删改查功能

学 时：3 学时

知识点：数据列表展示，权限点的设置

重点：角色和权限点的对应关系

难点：权限点的设置

讲解权限的作用，权限的增删改查的实现及权限树的前端代码的封装问题

任务指导

1. 新建权限页面
2. 完成权限后端的设计
3. 权限的前端的设计

任务实现