复习：

见思维导图

|  |
| --- |
| Model：模型 MVVM  Modal：模态对话框  Module：模块 |

1.Python中的模块和包

模块：Module，把所有的代码划分到不同的文件中，每个文件称为一个“模块”——便于分工协作；同时模块内数据只能在当前模块中使用，从而防止全局污染

|  |
| --- |
| CommonJS/Node.js中模块：  导出：module.exports = ...  导入：const x = require('./x') |
| ES6中的模块：  导出：export class Emp{ }  导入：import {Emp} from './x'  导出：export default class Emp{ }  导入：import Emp from './x' |
| Python中的模块：  导出：不需要，默认情况下所有成员都可以被其它模块导入  **导入形式1：**import m1 #m1就代表./m1.py文件模块  **导入形式2：**from m1 import age #age就是m1模块中的内部全局变量 |

注意：

①第一次导入某个模块会自动在内存中创建该模块对应的对象，会在内存中留下这份拷贝，后续如果还有针对同一个模块文件的导入，则不再执行该文件，直接从内存中取出之前的缓存对象——即每个模块对象都是“单例的(Singleton)”

②模块不需要导出内部成员——内部的全局变量就是模块的成员属性，全局函数就是模块的成员方法

③Python内部约定：启动模块的名称(\_\_name\_\_)为'\_\_main\_\_'；非启动模块的名称就是文件名称(去掉.py)。利用此特性，可以执行特定的单元测试代码：

if(\_\_name\_\_ == '\_\_main\_\_'):

#此处放置单元测试代码

④如果项目中的模块太多了，可以把多个模块文件放置在同一个目录下，这样的目录称为“package——包”；例如：“用户包”、“商品包”、“购物车包”...

假设有： user/login.py，则启动文件可以如下导入

导入包下的模块： import user.login as ul

导入包下的模块中的成员： from user.login import uname

⑤如果想在包中执行一些初始化代码，可以放在 **\_\_init\_\_.py** 的文件中——只要该包中任何模块被导入，都会执行且仅执行一次该初始化文件

|  |
| --- |
| CommonJS中有package的概念 —— 用于盛放多个模块文件  Python中也有package的概念；  ES6中没有package概念！！ |

练习：创建一个包shape，其中有两个模块：

circle(有变量r和方法getSize()/getPerimeter())

rectangle(有变量w/h和方法getSize()/getPerimeter())；

试着编写启动模块，调用包中这两个模块

2.Python官方提供的函数和对象

提示：官方提供的函数和对象都放置在特定的包或模块中；

手册：<https://docs.python.org/zh-cn/3/library/index.html>

**①datetime模块**

该模块提供了三个class：date/time/datetime

from datatime import date, time, datetime

**获取当前系统日期时间：**

dt = datetime.now()

**把日期时间格式化为特殊的字符串：**

s1 = dt.strftime( '%Y-%m-%d %H:%M:%S' )

------------------------------------------------

y = dt.year

m = dt.month

d = dt.day #不是date

h = dt.hour

mi = dt.minute

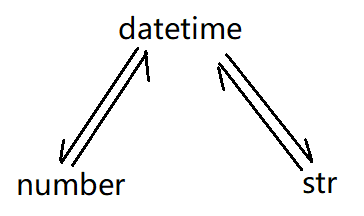
s = dt.second

**把日期时间转换为长整数(0表示1970-1-1 0:0:0)——称为时间戳：**

num = dt.timestamp( )

**把长整数转换为日期时间对象：**

dt = datetime.fromtimestamp( num )



**②math模块**

import math

**获取PI的值：**

math.pi

**上下取整：**

math.ceil( ) / math.floor( )

**三角函数：**

math.sin( deg )

注意：JS中还有Math.max()/min()/round()/abs()几个函数Python中是全局函数；JS中的Math.random()函数Python是一个独立的模块

**③random模块**

import random

**生成0~1间的随机小数：**

n = random.random( )

**生成min~max间的随机整数：**

n = random.randrange(min, max, step)

**从字符串数组中随机取出一个元素：**

e = random.choice( myList )

**随机打乱一个数组：**

random.shuffle( myList ) #直接修改原始列表

**④os.path模块**

提示：Python中没有类似Node.js中的\_\_dirname

**获取当前路径对应的绝对路径名：**

import os.path

p = os.path.abspath( './' )

**⑤json模块**

import json

**把dict对象转换为JSON格式的字符串——称为“序列化”：**

//JS：str = JSON.stringify( obj )

str = json.dumps( obj )

**把JSON格式的字符串解析为dict对象——称为“反序列化”：**

//JS: obj = JSON.parse( str )

obj = json.loads( str )

**⑥http.client模块**

模拟浏览器向指定的URL发起HTTP请求(GET/POST/...)

作用：爬虫程序、服务器端代理

import http.client

conn = http.client.HTTPConnection('www.codeboy.com') #创建连接

conn.request('GET', '/index.html') #发起请求

res = conn.getresponse( ) #得到响应消息

body = res.read( ) #读取响应主体

|  |
| --- |
| 面试题：跨域的解决方案？各自的利弊？  ①JSONP： 只能发起GET请求，且服务器端修改代码  ②CORS： 修改服务器端代码，生成特别响应消息头部  ③服务器端请求代理方案：让前端服务器代替客户端浏览器向后端服务器发请求 |
|  |

**⑦http.server模块**

|  |
| --- |
| 创建HTTP服务器的方法：  1)Node.js原生方案http 2)Node.js第三方框架：Express  3)Python原生方案http.server 4)Python第三方框架：Django |

3.Pip工具的使用——Python第三方包/模块下载工具

Node.js官方提供了很多模块，业界还有更多的第三方模块，默认托管在npmjs.com上，可以使用NPM（Node Package Manager）工具下载和维护第三方模块；

类似的，Python官方提供了很多模块，业界还有更多的第三方模块，默认托管在pypi.org上，可以使用PIP（Package Installer for Python）工具下载和维护第三方模块。

PIP常用的命令：

pip list #列出已经安装的所有模块

pip show 模块名 #显示指定模块的详情

pip install 模块名 #下载并安装指定模块

pip install --upgrade 模块名 #升级指定的模块

pip uninstall 模块名 #卸载指定模块

|  |
| --- |
| 提示：PIP默认安装的模块全部安装在 c:\users\web\appdata\local\programs\python\python38-32\lib\site-packages |

演示：第三方模块的使用——MySQL连接器模块

通过查找仓库，可用的模块有：mysql-connector-python —— 该模块的作者是mysql.com。使用该模块操作MySQL数据库的步骤：

①启动本地的MySQL数据库服务器

②下载并安装第三方mysql连接模块

pip install mysql-connector-python

使用pip list查看所安装的模块以验证是否安装成功

③导入mysql连接器模块，连接到MySQL服务器上

import mysql.connector

cnx = mysql.connector.connect(

user='root',

password='',

host='127.0.0.1',

port='3306',

database='xz'

)

④向MySQL连接发送“增删改查”SQL命令

阅读手册中的示例代码继续完成

课后任务：

(1)阅读MySQL连接器的说明手册，继续完成使用第三方模块执行SQL语句的操作：

<https://dev.mysql.com/doc/connector-python/en/connector-python-examples.html>

(2)将电脑连网，使用pip下载第三方Python框架：Django

pip install django

并开始自学该框架的使用方法，

手册：<https://docs.djangoproject.com/zh-hans/2.2/>