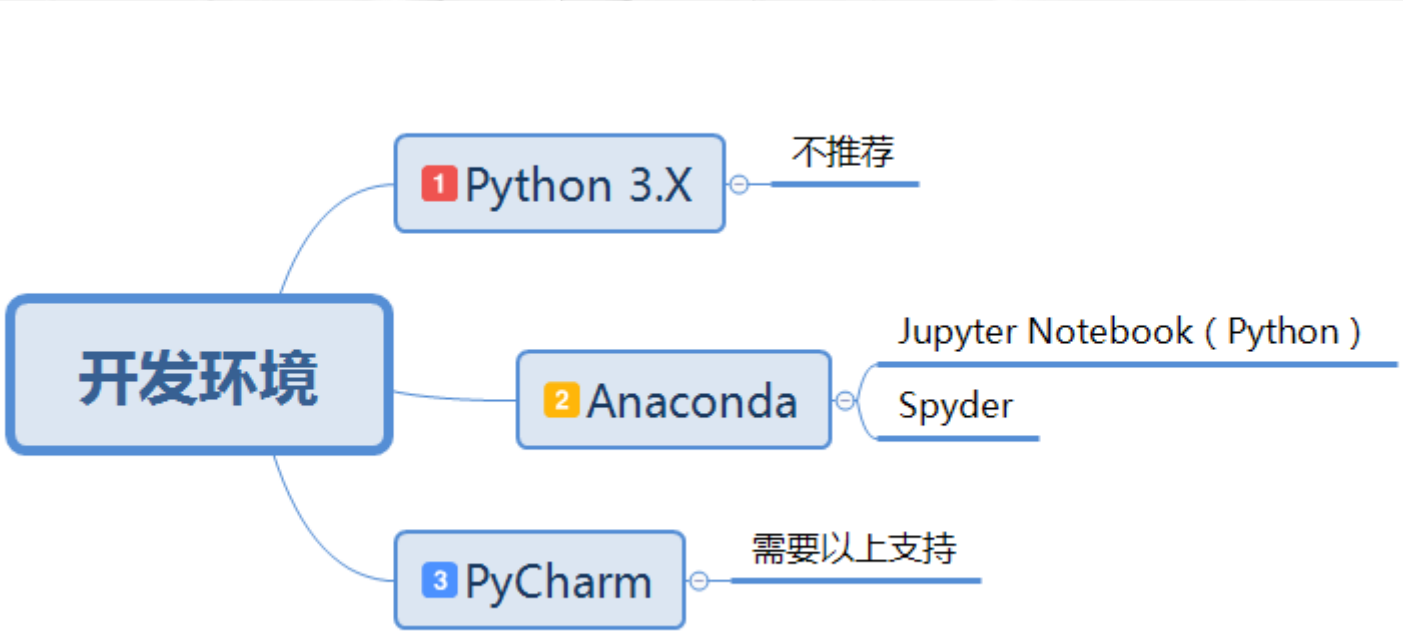
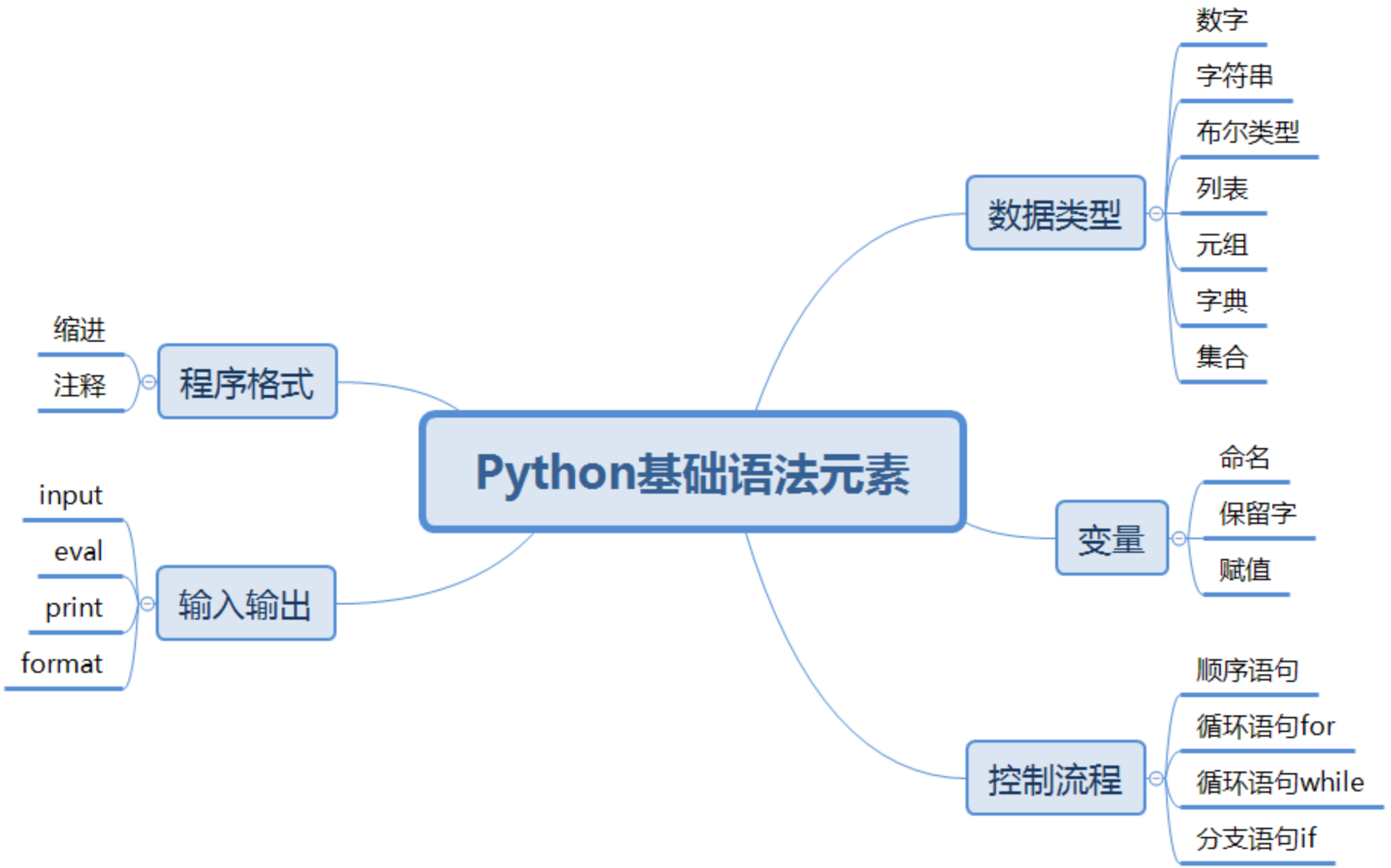
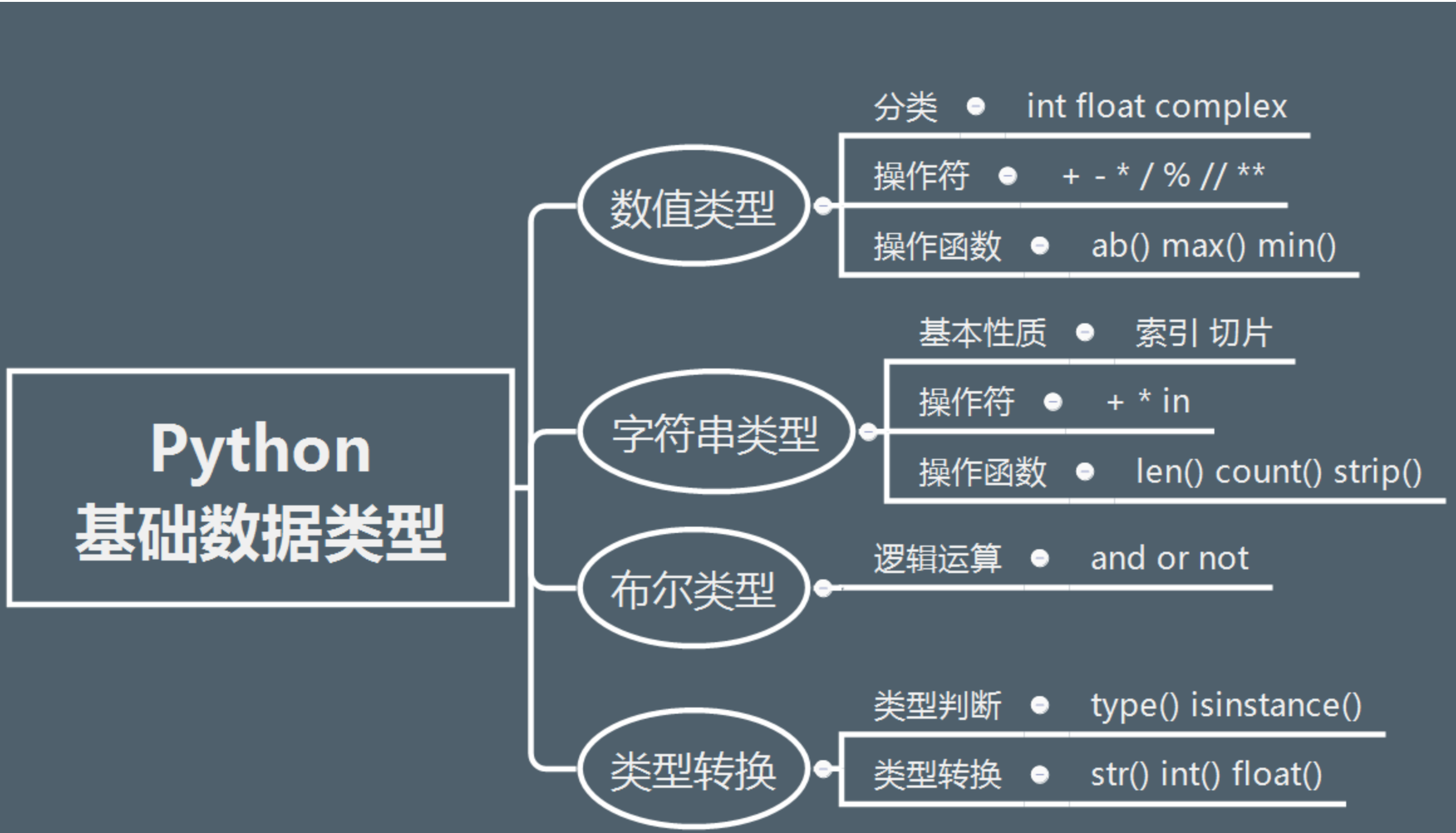
数据分析与可视化技术
Data Analysis for Infomatics

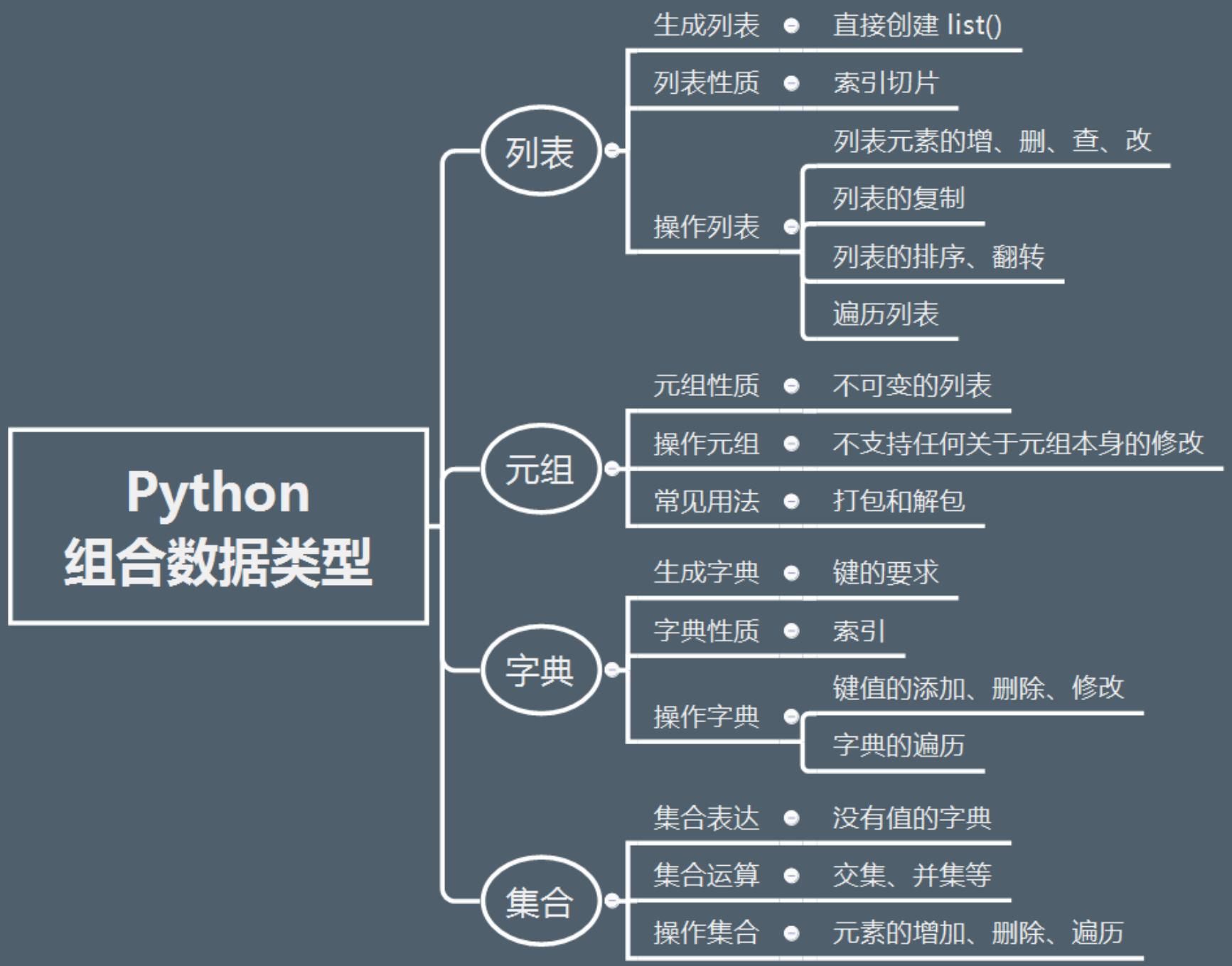
第一讲 Python提升探索, Numpy & Pandas

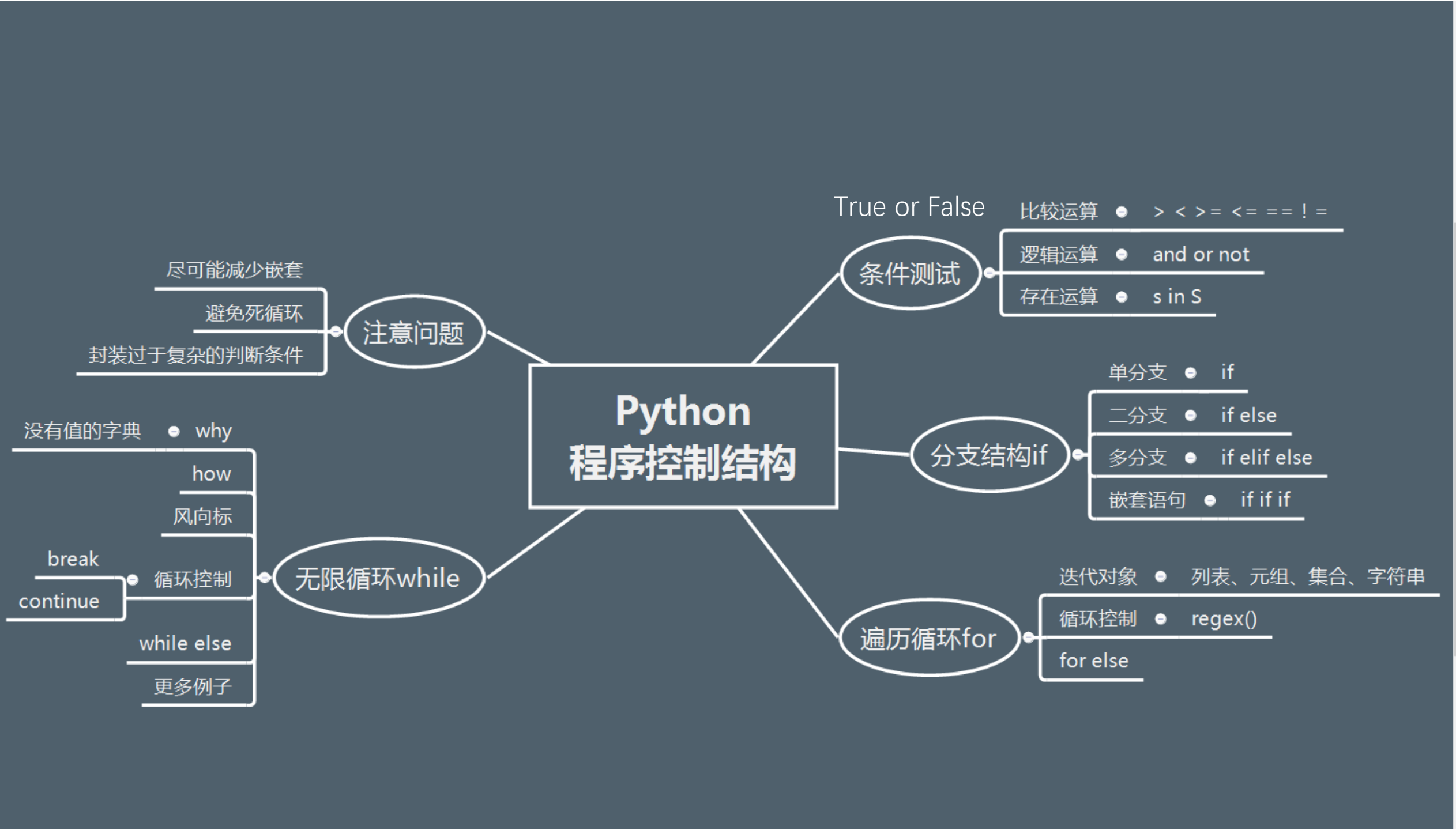
刘子瑜 副研究馆员
liuziyu@cau.edu.cn 中国农业大学图书馆情报研究中心
https://github.com/CAU-LiuZiyu/Data_Analysis_for_Infomatics

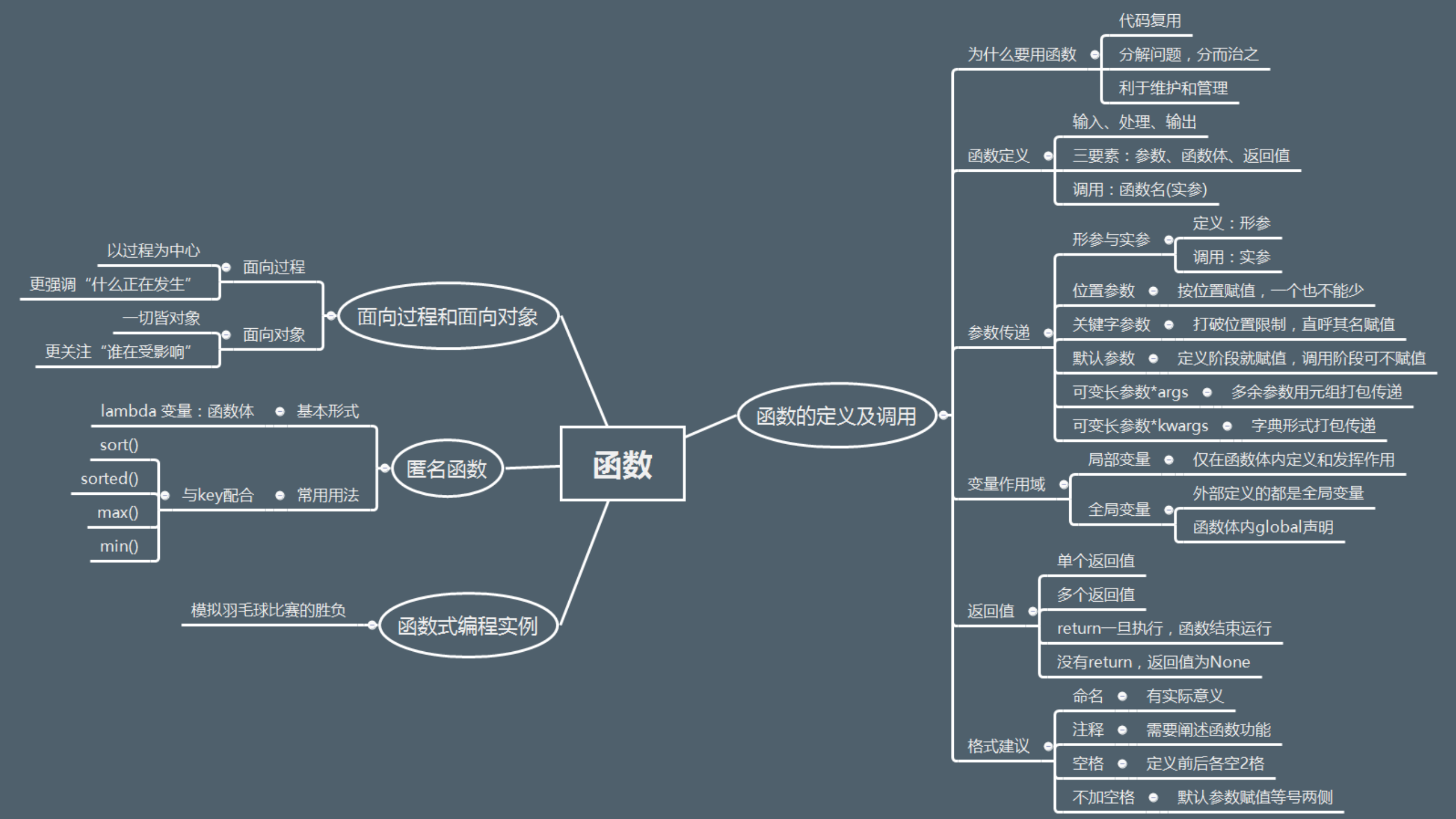

开发环境
1 Python 3.X
不推荐
2 Anaconda
Jupyter Notebook (Python)
Spyder
3 PyCharm
需要以上支持

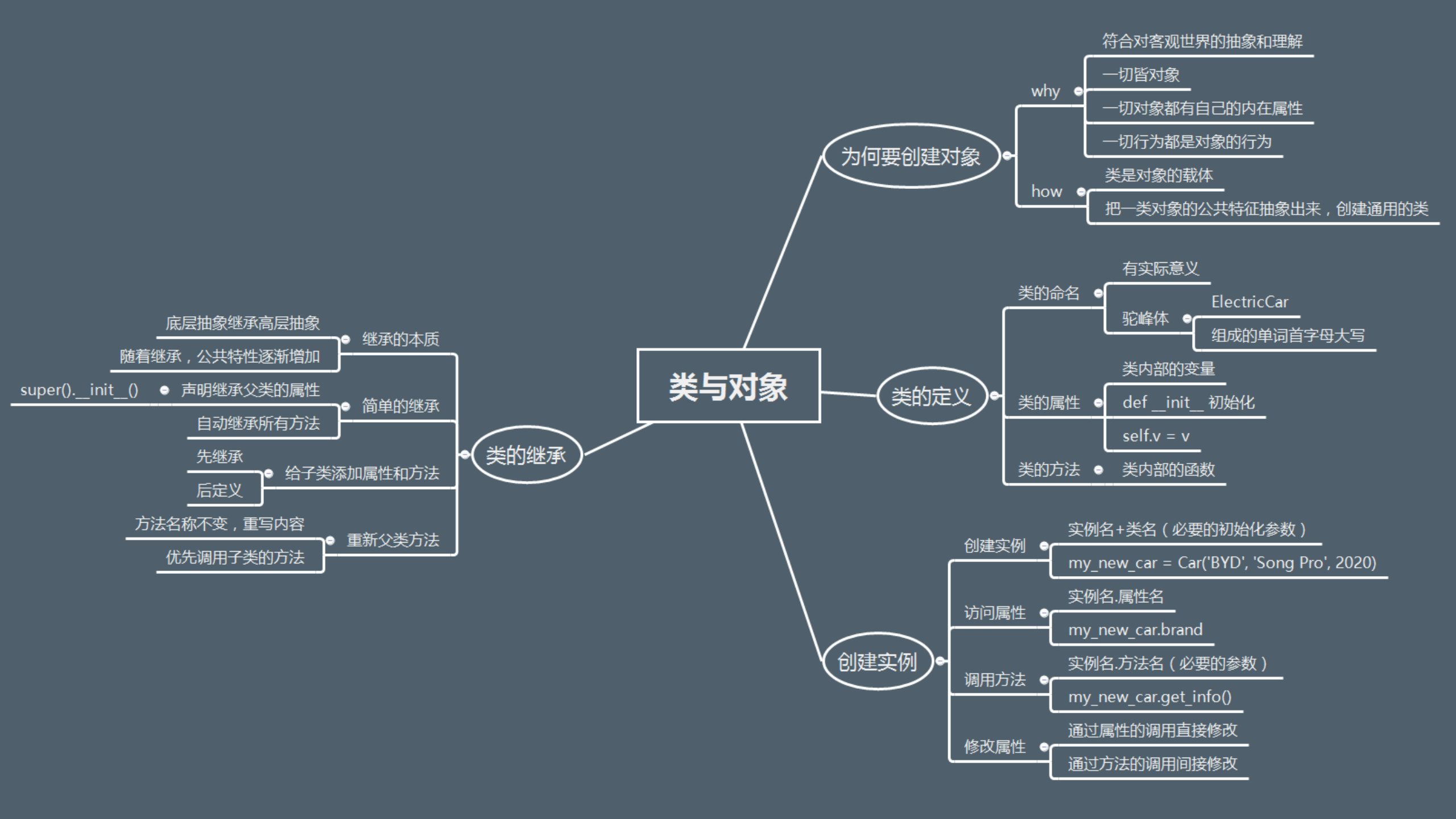

Python基础语法元素
数据类型
数字
字符串
布尔类型
列表
元组
字典
集合
变量
命名
保留字
赋值
控制流程
顺序语句
循环语句for
循环语句while
分支语句if
程序格式
缩进
注释
输入输出
input
eval
print
format

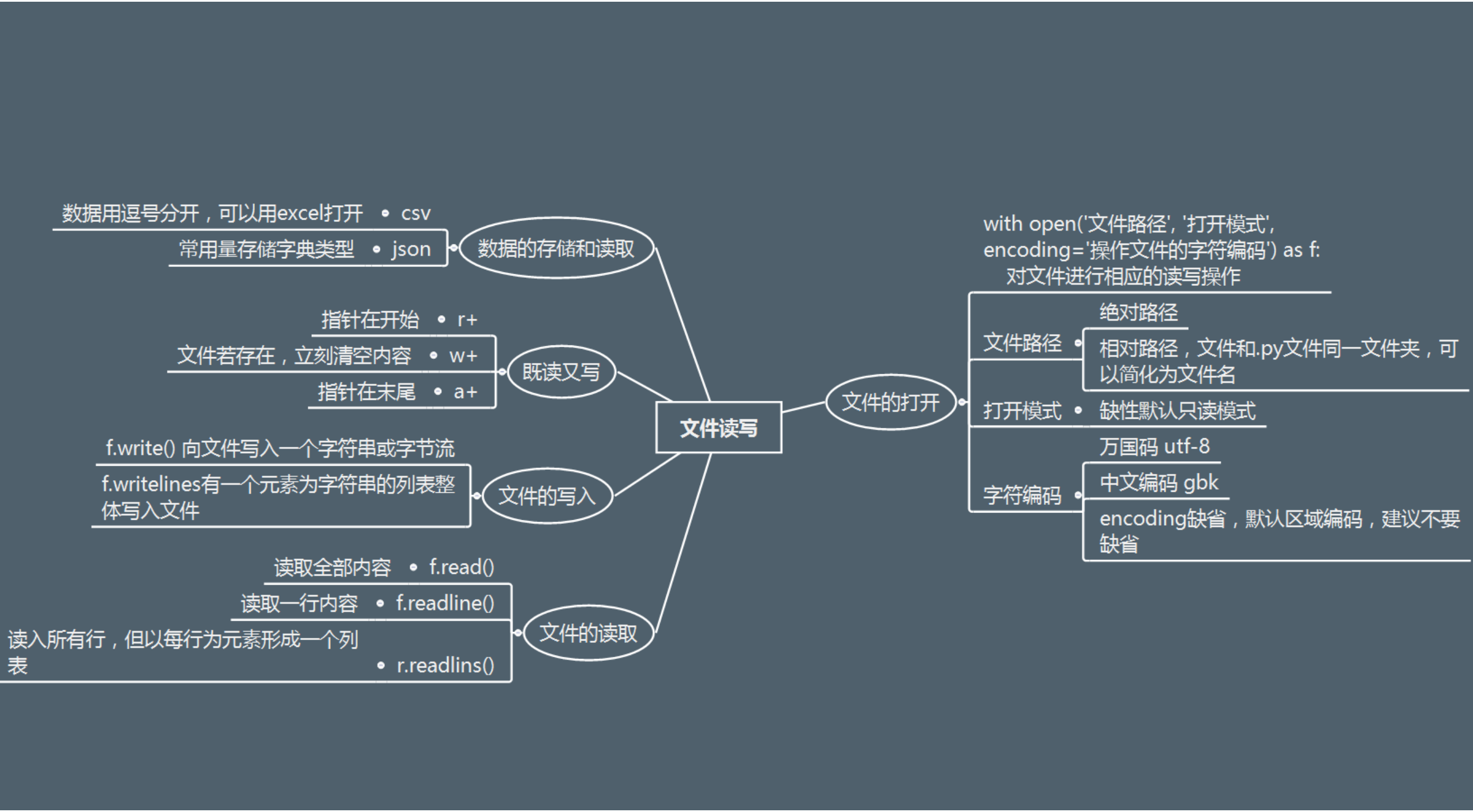

Python
基础数据类型
数值类型
分类 int float complex
操作符 + - * / % // **
操作函数 ab() max() min()
字符串类型
基本性质 索引 切片
操作符 + * in
操作函数 len() count() strip()
布尔类型
逻辑运算 and or not
类型转换
类型判断 type() isinstance()
类型转换 str() int() float()

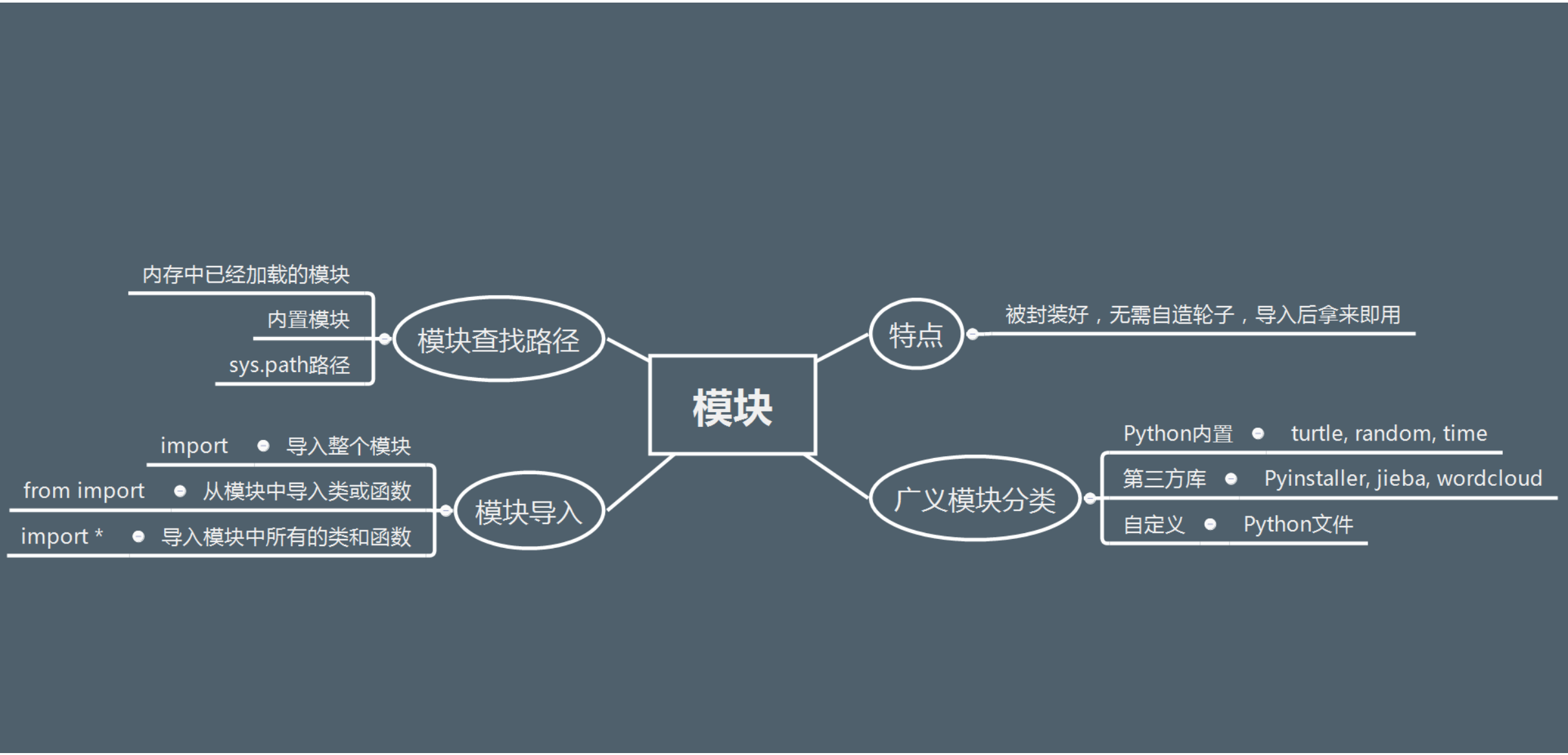

Python
组合数据类型
列表
生成列表
直接创建 list()
列表性质
索引切片
操作列表
列表元素的增、删、查、改
列表的复制
列表的排序、翻转
遍历列表
元组
元组性质
不可变的列表
操作元组
不支持任何关于元组本身的修改
常见用法
打包和解包
字典
生成字典
键的要求
字典性质
索引
操作字典
键值的添加、删除、修改
字典的遍历
集合
集合表达
没有值的字典
集合运算
交集、并集等
操作集合
元素的增加、删除、遍历

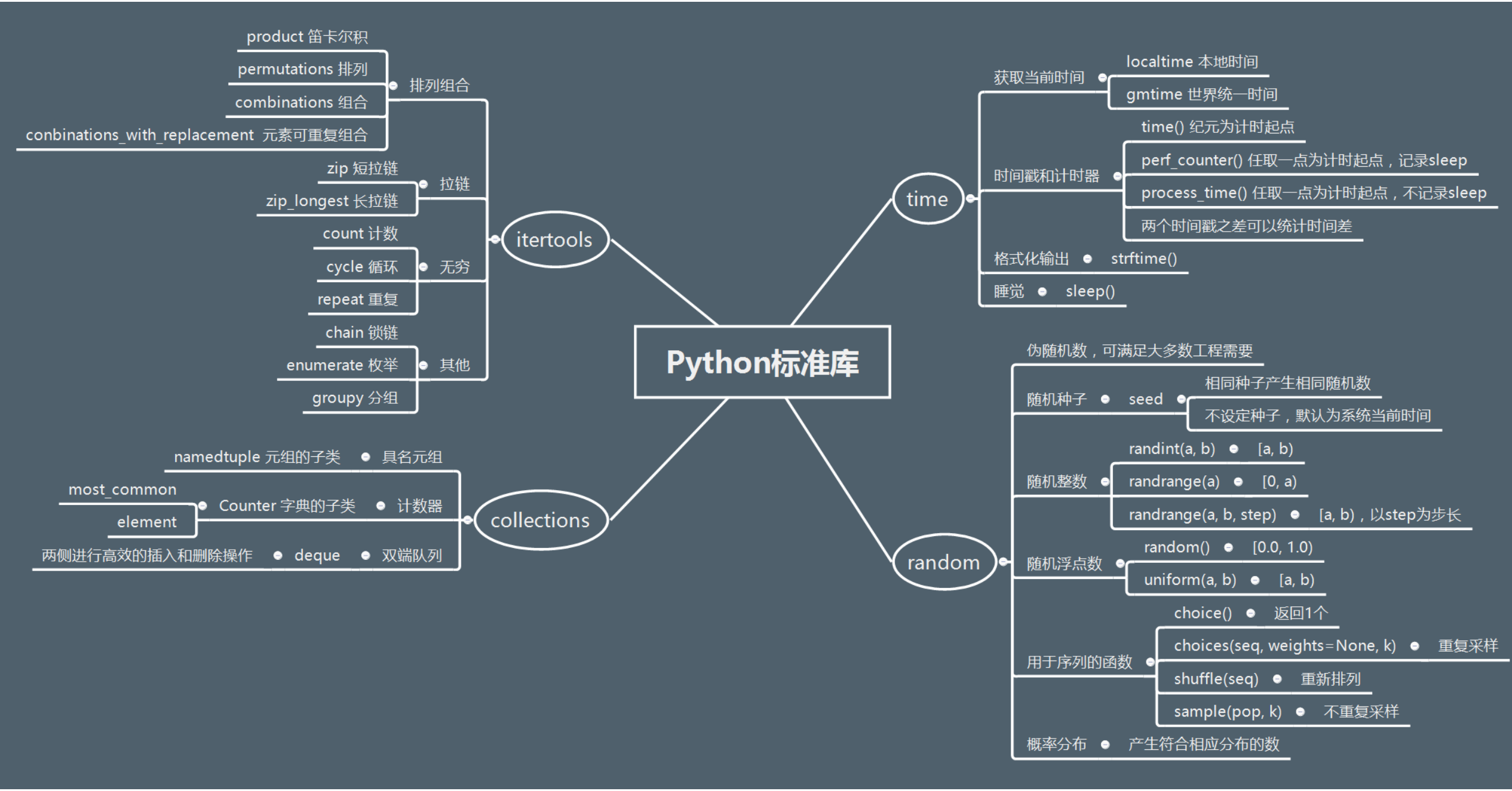

Python
程序控制结构
条件测试
True or False
比较运算
> < >= <= == ! =
逻辑运算
and or not
存在运算
s in S
分支结构if
单分支
if
二分支
if else
多分支
if elif else
嵌套语句
if if if
遍历循环for
迭代对象
列表、元组、集合、字符串
循环控制
regex()
for else
无限循环while
没有值的字典
why
how
风向标
循环控制
break
continue
while else
更多例子
注意问题
尽可能减少嵌套
避免死循环
封装过于复杂的判断条件

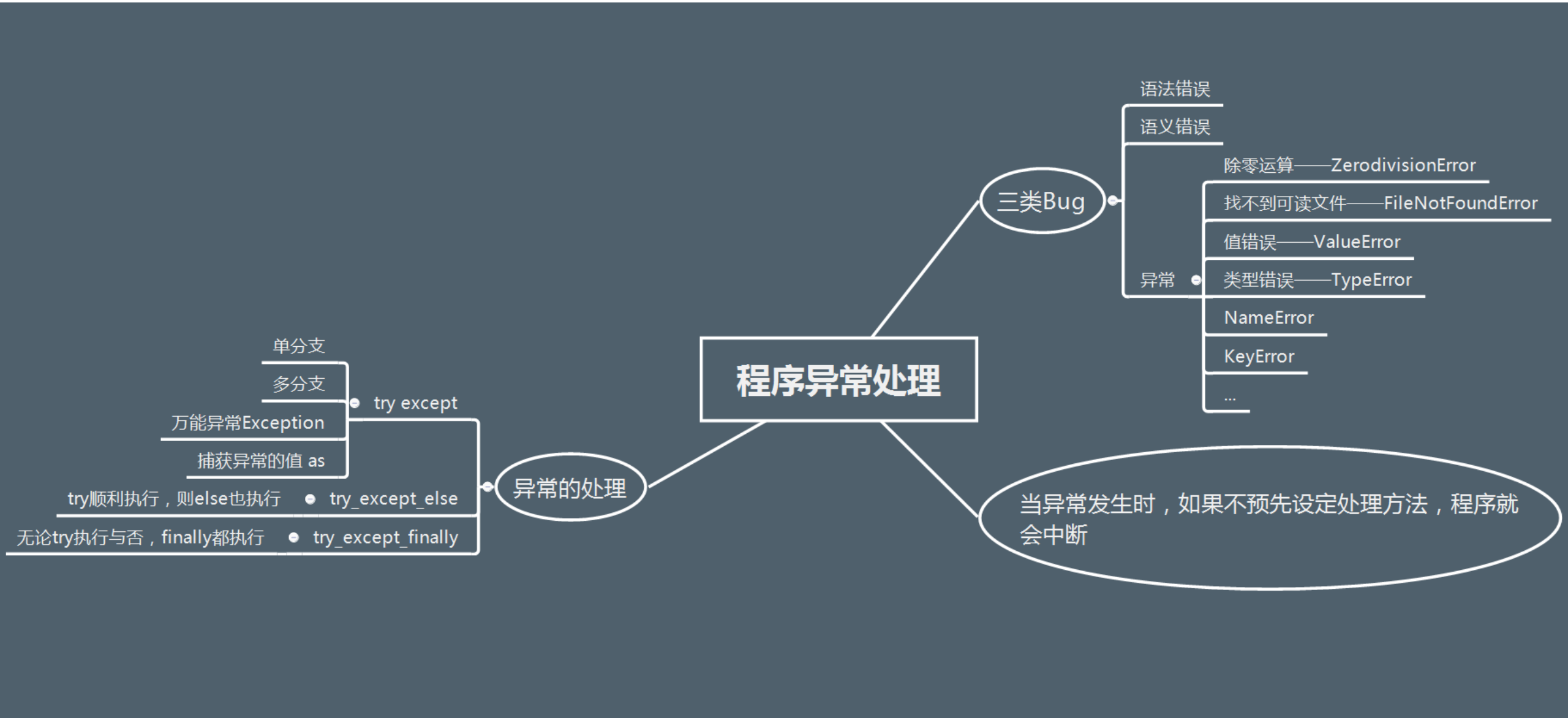

函数
函数的定义及调用
为什么要用函数
代码复用
分解问题，分而治之
利于维护和管理
函数定义
输入、处理、输出
三要素：参数、函数体、返回值
调用：函数名(实参)
参数传递
形参与实参
定义：形参
调用：实参
位置参数
按位置赋值，一个也不能少
关键字参数
打破位置限制，直呼其名赋值
默认参数
定义阶段就赋值，调用阶段可不赋值
可变长参数*args
多余参数用元组打包传递
可变长参数*kwargs
字典形式打包传递
变量作用域
局部变量
仅在函数体内定义和发挥作用
全局变量
外部定义的都是全局变量
函数体内global声明
返回值
单个返回值
多个返回值
return一旦执行，函数结束运行
没有return，返回值为None
格式建议
命名
有实际意义
注释
需要阐述函数功能
空格
定义前后各空2格
不加空格
默认参数赋值等号两侧
面向过程和面向对象
面向过程
以过程为中心
更强调“什么正在发生”
面向对象
一切皆对象
更关注“谁在受影响”
匿名函数
lambda 变量：函数体
基本形式
与key配合
常用用法
sort()
sorted()
max()
min()
函数式编程实例
模拟羽毛球比赛的胜负

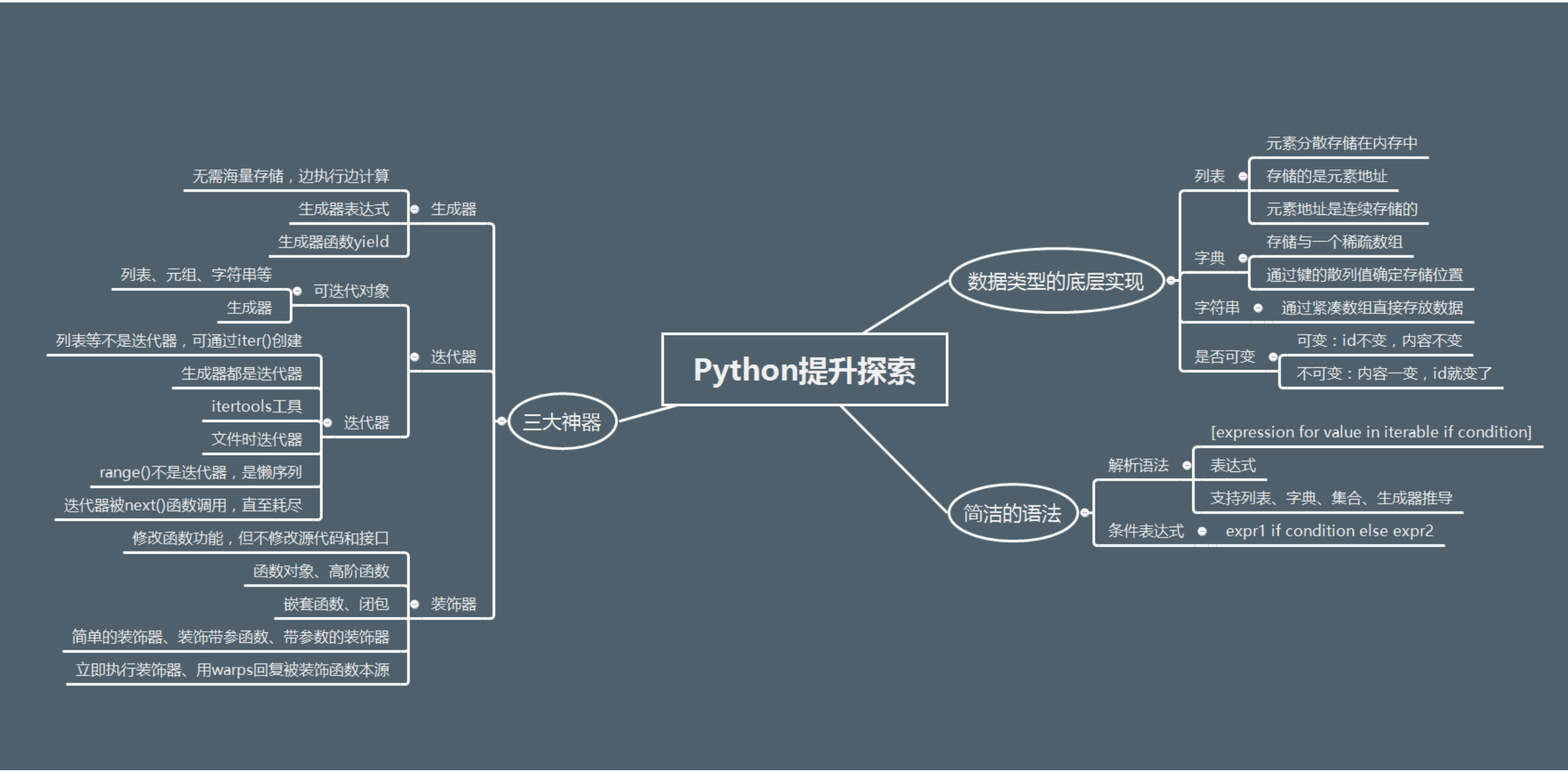

类与对象
为何要创建对象
why
符合对客观世界的抽象和理解
一切皆对象
一切对象都有自己的内在属性
一切行为都是对象的行为
how
类是对象的载体
把一类对象的公共特征抽象出来，创建通用的类
类的定义
类的命名
有实际意义
驼峰体
ElectricCar
组成的单词首字母大写
类的属性
类内部的变量
def __init__ 初始化
self.v = v
类的方法
类内部的函数
创建实例
创建实例
实例名+类名（必要的初始化参数）
my_new_car = Car('BYD', 'Song Pro', 2020)
访问属性
实例名.属性名
my_new_car.brand
调用方法
实例名.方法名（必要的参数）
my_new_car.get_info()
修改属性
通过属性的调用直接修改
通过方法的调用间接修改
类的继承
继承的本质
底层抽象继承高层抽象
随着继承，公共特性逐渐增加
简单的继承
super().__init__()
声明继承父类的属性
自动继承所有方法
给子类添加属性和方法
先继承
后定义
重新父类方法
方法名称不变，重写内容
优先调用子类的方法

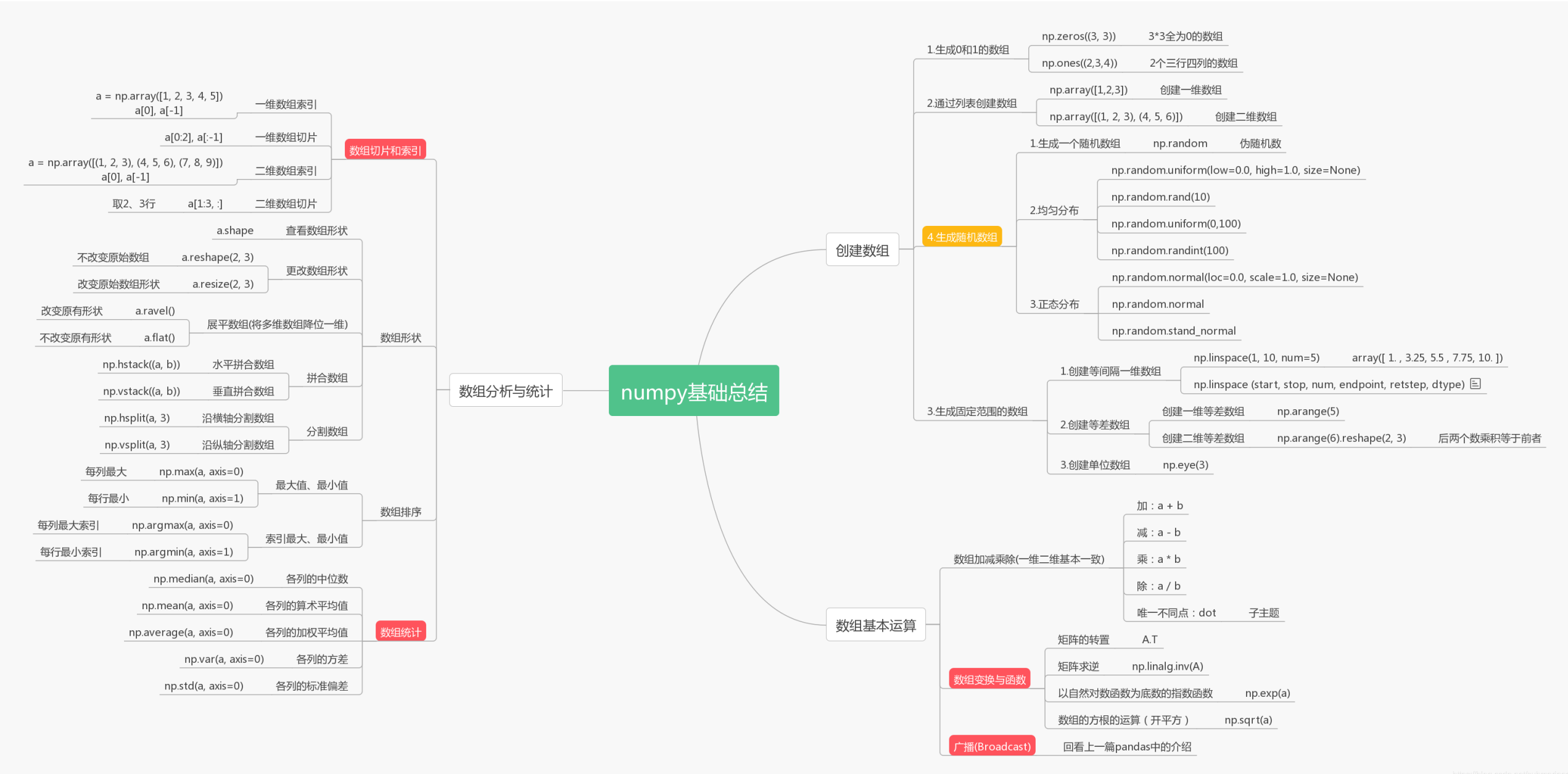

文件读写
文件的打开
with open('文件路径', '打开模式', encoding='操作文件的字符编码') as f:
对文件进行相应的读写操作
文件路径
绝对路径
相对路径，文件和.py文件同一文件夹，可以简化为文件名
打开模式
缺性默认只读模式
字符编码
万国码 utf-8
中文编码 gbk
encoding缺省，默认区域编码，建议不要缺省
数据的存储和读取
数据用逗号分开，可以用excel打开
csv
常用量存储字典类型
json
既读又写
指针在开始
r+
文件若存在，立刻清空内容
w+
指针在末尾
a+
文件的写入
f.write() 向文件写入一个字符串或字节流
f.writelines有一个元素为字符串的列表整体写入文件
文件的读取
读取全部内容
f.read()
读取一行内容
f.readline()
读入所有行，但以每行为元素形成一个列表
r.readlins()


模块
模块查找路径
内存中已经加载的模块
内置模块
sys.path路径
模块导入
import 导入整个模块
from import 从模块中导入类或函数
import * 导入模块中所有的类和函数
特点
被封装好，无需自造轮子，导入后拿来即用
广义模块分类
Python内置 turtle, random, time
第三方库 Pyinstaller, jieba, wordcloud
自定义 Python文件

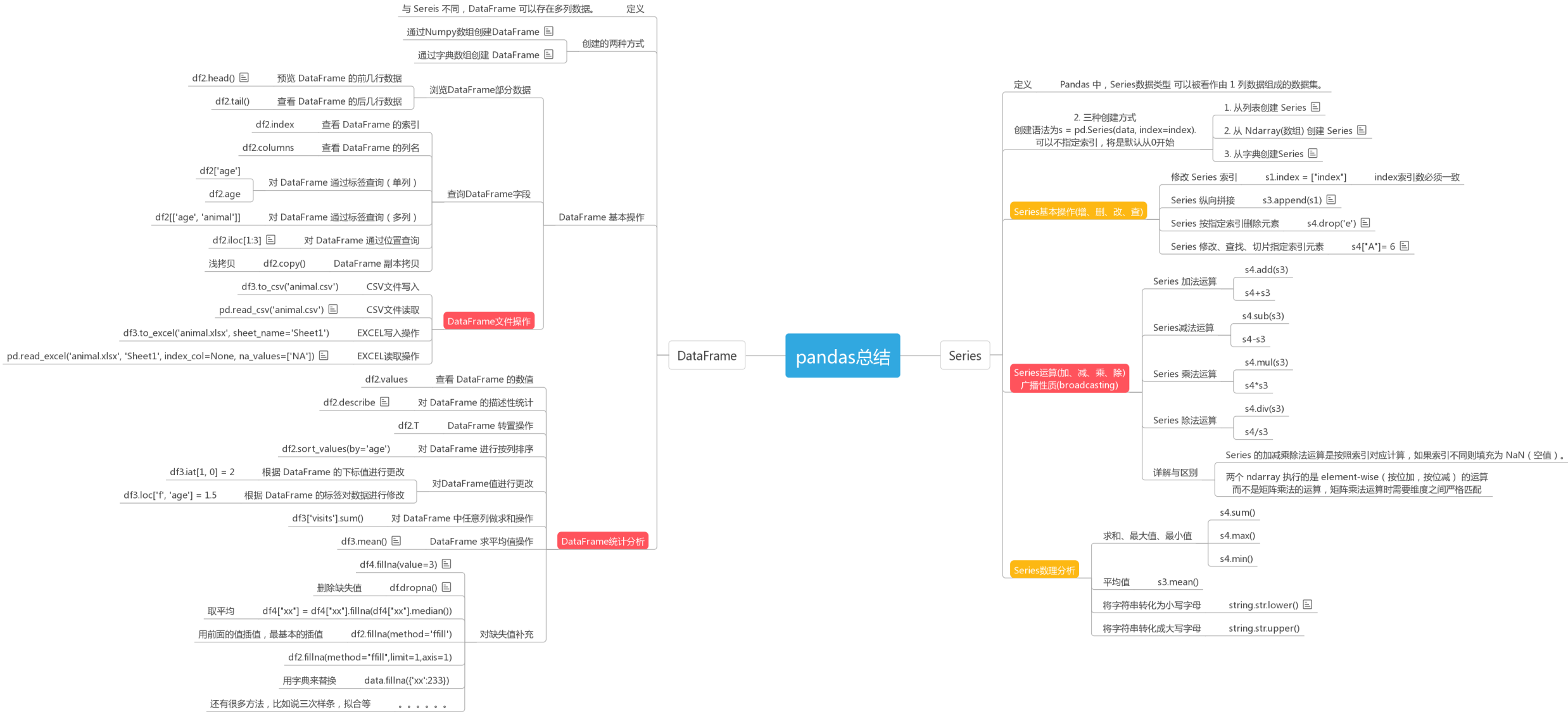

Python标准库
itertools
排列组合
product 笛卡尔积
permutations 排列
combinations 组合
conbinations_with_replacement 元素可重复组合
拉链
zip 短拉链
zip_longest 长拉链
无穷
count 计数
cycle 循环
repeat 重复
其他
chain 锁链
enumerate 枚举
groupy 分组
collections
具名元组
namedtuple 元组的子类
计数器
Counter 字典的子类
most_common
element
双端队列
deque
两侧进行高效的插入和删除操作
time
获取当前时间
localtime 本地时间
gmtime 世界统一时间
时间戳和计时器
time() 纪元为计时起点
perf_counter() 任取一点为计时起点，记录sleep
process_time() 任取一点为计时起点，不记录sleep
两个时间戳之差可以统计时间差
格式化输出
strftime()
睡觉
sleep()
random
伪随机数，可满足大多数工程需要
随机种子
seed
相同种子产生相同随机数
不设定种子，默认为系统当前时间
随机整数
randint(a, b)
[a, b]
randrange(a)
[0, a)
randrange(a, b, step)
[a, b)，以step为步长
随机浮点数
random()
[0.0, 1.0)
uniform(a, b)
[a, b)
用于序列的函数
choice()
返回1个
choices(seq, weights=None, k)
重复采样
shuffle(seq)
重新排列
sample(pop, k)
不重复采样
概率分布
产生符合相应分布的数


程序异常处理
三类Bug
语法错误
语义错误
异常
除零运算——ZerodivisionError
找不到可读文件——FileNotFoundError
值错误——ValueError
类型错误——TypeError
NameError
KeyError
...
异常的处理
try except
单分支
多分支
万能异常Exception
捕获异常的值 as
try顺利执行，则else也执行
try_except_else
无论try执行与否，finally都执行
try_except_finally
当异常发生时，如果不预先设定处理方法，程序就会中断


Python提升探索
三大神器
生成器
无需海量存储，边执行边计算
生成器表达式
生成器函数yield
迭代器
可迭代对象
列表、元组、字符串等
生成器
迭代器
列表等不是迭代器，可通过iter()创建
生成器都是迭代器
itertools工具
文件时迭代器
range()不是迭代器，是懒序列
迭代器被next()函数调用，直至耗尽
装饰器
修改函数功能，但不修改源代码和接口
函数对象、高阶函数
嵌套函数、闭包
简单的装饰器、装饰带参函数、带参数的装饰器
立即执行装饰器、用warps回复被装饰函数本源
数据类型的底层实现
列表
元素分散存储在内存中
存储的是元素地址
元素地址是连续存储的
字典
存储与一个稀疏数组
通过键的散列值确定存储位置
字符串
通过紧凑数组直接存放数据
是否可变
可变：id不变，内容不变
不可变：内容一变，id就变了
简洁的语法
解析语法
[expression for value in iterable if condition]
表达式
支持列表、字典、集合、生成器推导
条件表达式
expr1 if condition else expr2



https://blog.csdn.net/submarineas/article/details/85041240



https://blog.csdn.net/submarineas/article/details/84800227