python蓝桥杯基础知识总结：

1.模块方面

1.1statistics

```
import statistics
list0=[2,3,2,3,5,6,7,8]
print(statistics.mean(list0))
print(statistics.median(list0))
print(statistics.median_low(list0))
print(statistics.median_high(list0))
print(statistics.mode(list0))
print(statistics.pvariance(list0))
#1、statistics.mean()  求算术平均值
#2、statistics.median() 计算数据的中位数，如果有两个中位数，则返回其平均值
#    statistics.median_low() 数据中的低中位数
#    statistics.median_high() 数据中的高中位数
# 3、statistics.mode()  计算众数
# 4、statistics.pvariance() 计算数据的总体方差
```

1.2collections

```
import collections
q = collections.deque([1, 2, 3, 4])
q.rotate(1)
print(q)  # [4, 1, 2, 3]
q.rotate(1)
print(q)  # [3, 4, 1, 2]
##############################
from collections import deque
q=deque([1,2,3,4])
temp=[2,5,6]
q.extendleft(temp)
print(list(q))
# deque([6, 5, 2, 1, 2, 3, 4])
# [6, 5, 2, 1, 2, 3, 4]
```

1.3calendar

```
import calendar
print(calendar.isleap(2022))
print(calendar.leapdays(2000,2022))
# False
# 6
```

1.4datetime

```
import datetime
bt = datetime.date(2000,11,6)
print(bt)
# 2000-11-06
a = datetime.timedelta(days=100)
datetime.timedelta(days=100) #weeks / hours
b = a + bt
datetime.date(2001, 2, 14)

# bt.weekday()：返回weekday，如果是星期一，返回0；如果是星期2，返回1，以此类推；
# bt.isoweekday()：返回weekday，如果是星期一，返回1；如果是星期2，返回2，以此类推；
```

1.5math

```
# math.ceil(x)：返回不小于 x 的最小整数。
# math.floor(x)：返回不大于 x 的最大整数。
# math.trunc(x)：返回 x 的整数部分。
# math.modf(x)：返回 x 的小数部分和整数部分，以元组形式返回。
# math.fabs(x)：返回 x 的绝对值。
# math.factorial(x)：返回 x 的阶乘。
# math.gcd(a, b)：返回 a 和 b 的最大公约数。
```

1.6itertools

```
import itertools
   # 1)product
for i in itertools.product('ab', 'cd'):
    print(i)
# >>>('a', 'c')
#    ('a', 'd')
#    ('b', 'c')
#    ('b', 'd')

  # 2)permutations
    list(itertools.permutations('ABC'))
    # >>>[('A', 'B', 'C'), ('A', 'C', 'B'), ('B', 'A', 'C'), ('B', 'C', 'A'),
('C', 'A', 'B'), ('C', 'B', 'A')]
    list(itertools.permutations('ABC', 2))
    # >>>[('A', 'B'), ('A', 'C'), ('B', 'A'), ('B', 'C'), ('C', 'A'), ('C', 'B')]

    # 3)combinations
    list(itertools.combinations("ABC",2))
    # >>>[('A', 'B'), ('A', 'C'), ('B', 'C')]

    # 4)accumulate 可指定函数，默认为求和
    list(itertools.accumulate([0,1,0,1,1,2,3,5]))
    # >>>[0, 1, 1, 2, 3, 5, 8, 13]
    list(itertools.accumulate([0,1,0,1,1,2,3,5],max))
    # >>>[0, 1, 1, 1, 1, 2, 3, 5]
```

1.7bisect

```
# bisect_left(a, x, lo=0, hi=len(a))
# bisect_right(a, x, lo=0, hi=len(a))
# insort_left(a, x, lo=0, hi=len(a))
# insort_right(a, x, lo=0, hi=len(a))
```

2.简单知识点

2.1functions

```
# 返回单个字母的特定值，可以先返回其Unicode码
# ord("a")
# >> > 97

# 千位分隔符
# print({:,}.format(n).replace(",","."))
# n = 123456
# >>>123.456

# 求a和b最大公约数
#   gcd(a,b)

#  二进制
# bin() 返回一个整数 int 或者长整数 long int 的二进制表示。

# 二进制转十进制 int(“二进制字符串”,2)
# 十进制转二进制 bin(整数)[2:]

# 幂函数
# pow(a,b)

# enumerate() 函数
# 用于将一个可遍历的数据对象(如列表、元组或字符串)组合为一个索引序列，
# 同时列出数据和数据下标，一般用在 for 循环当中。
# max()和min()，字符串是可以比较的，
# 按照ascII值排 abb， aba，abac，最大为abb，最小为aba
```

2.2set

```
# 并：|  交：  &   差：  -  对称差集：  ^

# 返回一个新集合，该集合的元素既包含在集合 x 又包含在集合 y 中
x=set([2,5,6,3,4])
y=set([2,6,9,8,5])
print(x.intersection(y))
# {2, 5, 6}

# symmetric_difference(other_set) 或 ^：返回两个集合的对称差集（即只属于其中一个集合的元素）。
s1 = {1, 2, 3}
```

```
s2 = {3, 4, 5}
s3 = s1.symmetric_difference(s2)
print(s3)  # 输出：{1, 2, 4, 5}


# discard(element)：移除集合中的一个元素，如果元素不存在不会引发错误。
s = {1, 2, 3}
s.discard(4)
print(s)  # 输出：{1, 2, 3}

# clear()：清空集合中的所有元素。
s = {1, 2, 3}
s.clear()
print(s)  # 输出：set()

# set函数可以进行比较
a = set('qwertyuiop')
b = set("reagdagd")
print(a)
print(b)
print(a<b)
# {'t', 'r', 'p', 'i', 'w', 'o', 'e', 'y', 'u', 'q'}
# {'r', 'a', 'e', 'g', 'd'}
# True
```

2.3list

```
# zip函数
a = [1,2,3]
b = [4,5,6]
list(zip(a,b))
# >>>[(1,4),(2,5),(3,6)]
strs = ["flower","flow","flight"]
# list(zip(*strs))
print(list(zip(*strs)))
# [('f', 'f', 'f'), ('l', 'l', 'l'), ('o', 'o', 'i'), ('w', 'w', 'g')]

# 将两个列表交叉合并
from itertools import chain
a=[1,2,3,4,5,6]
ans=[8,9,10,22]
sss=list(chain.from_iterable(zip(ans, a)))
# [8, 1, 9, 2, 10, 3, 22, 4]

# 获取二维数组每列元素
matrix = [[3,7,8],[9,11,13],[15,16,17]]
colmin = [i for i in zip(*matrix)]
# >>>[(3, 9, 15), (7, 11, 16), (8, 13, 17)]

# bisect.bisect_left(list , i) 查找该数值将会插入的位置并返回
# 字符串.rfind() 查找字符最后一次出现的位置，没有则返回-1
```

```
# 按照某种规律进行排序
l = [[5,10],[2,5],[4,7],[3,9]]
l = sorted(l, key=lambda x:-x[1])
# >>>[[5, 10], [3, 9], [4, 7], [2, 5]]
l = ["adss","a","sad","sd"]
l = sorted(l,key=lambda x:len(x))
# >>>['a', 'sd', 'sad', 'adss']

# 找出给定列表的所有子集
nums=[1,2,3,4]
l = [[]]
for x in nums:
    l.extend([i + [x] for i in l])

# 对序列中的元素进行累计 eg：求所有元素的异或总和
from functools import reduce
j = [1,2,3,4,5]
reduce(lambda x,y:x ^ y,j)
```

2.4string

```
# find在字符串中查找子串，如果找到，返回字串的第一个字符的索引，找不到返回-1
a='cbacbis'
b='bi'
print(a.find(b))
# 4

# 字符串.rfind() 查找字符最后一次出现的位置，没有则返回-1

# replace将指定字符串替换为另一个字符串
   #str.replace(指定字符，目标字符，不超过的次数)
```

2.5dict

```
# 字母异位词分组
strs = ["eat", "tea", "tan", "ate", "nat", "bat"]
res = []
dic = {}
for s in strs:
    keys = "".join(sorted(s))
    if keys not in dic:
        dic[keys] = [s]
    else:
        dic[keys].append(s)
print(list(dic.values()))
# >>>[['eat', 'tea', 'ate'], ['tan', 'nat'], ['bat']]

 # 字典的键值对反过来
b = {v:k for k,v in d.items()}
for i in d:
    if ans[i] == a:
        l.append(i)
```

```python
# 字典根据值的大小进行排序
d = sorted(d.items(), key=lambda item:item[1])
```