Python期末大纲

1、什么是匿名函数，以及其意义，使用场景

在Python的开发过程中，经常使用某一类函数，仅仅需要使用一次，没有必要单独定义出来，使用匿名函数就可以了，def函数重点去处理负责的业务。

2、raw字符串什么意义

告诉解释器忽略字符串中的转义字符，即在处理包含大量反斜杠的字符串时，使用原始字符串可以避免手动添加额外的转义字符，使得代码更加清晰和易读。

3、Python核心数据类型

数组、字符串、元组、字典、列表

4、字符串的常见操作

参考Day03、Day04、Day05

5、列表的切片和列表的切片赋值

a.列表的切片：

列表[:]，列表[::]

列表的切片返回一个新的列表，原列表内容不变

b.列表的切片赋值：

针对原列表进行的操作，改变切片区域的序列，切片的同时进行赋值，赋值的内容必须是可迭代对象。

注意：如果切片的步长大于1的时候，等号右边的可迭代对象的元素的个数要相同。

6、列表和元组的区别

a.元组的元素不可以修改，而列表可以

b.元组使用小括号，列表使用中括号

7、字典的存储特性

a.字典是一种可变的容器，可以存储任意类型的数据

b.字典中的每一个数据都用键(key)进行索引，而不是像序列那样使用下标

c.字典中的元素没有先后顺序，字典的存储是无序的

d.字典的数据都是以key-value键值对的方式进行的存储映射

e.字典的key值不能重复，且只能是不可变类型作为字典的键

8、函数的意义

a.结构化和模块化 函数允许将代码分割为小块，每个函数负责特定的任务。这种结构化编程方式使得代码更易于理解、维护和扩展，同时也促进了代码的模块化，可以将函数组合成更大的功能模块。

b.代码重用 函数可以被重复调用，避免了重复编写相同功能的代码。这样做不仅提高了代码的复用性，也减少了代码量，使得程序更加简洁和可维护。

c.提高可读性 通过函数命名和注释，可以更清晰地表达代码的意图和功能。函数提供了一种高层次的抽象，使得代码更易于阅读和理解。

d.参数传递 函数可以接受参数，这样可以将数据传递给函数进行处理。参数允许函数在不同的上下文中执行相同的操作，增加了函数的灵活性和通用性。

e.减少错误和调试 将代码分解为函数可以减少代码的复杂性，每个函数专注于特定的任务，减少了出错的可能性。当出现问题时，函数的模块化结构也使得调试更加容易。

f.支持高级特性 函数也是许多Python高级特性的基础，例如递归、闭包、装饰器等。这些特性增强了函数的灵活性和功能。

9、函数的调用时，参数的传递规则

a.位置传参

实参(实际参数)和形参的对应关系按照位置依次进行匹配

b.序列传参

在函数的调用过程中，使用\*将序列拆散后再按照位置传参的方式进行参数传递

c.关键字传参

传参的时候，使用形参的关键字进行传递；形参进行匹配时，其顺序可以不受约束，只要保证，每一个形参都被匹配到就可以。

d.字典关键字传参

实参为字典，使用\*\*拆解后，进行关键字传参；字典的key要和形参保持一致；字典的key要是字符串

10、函数定义时，形参的定义规则

A.位置形参

def fun(形参1，形参2…)

Pass

B.星号元组形参

def fun(\*args)

pass

以元组的形式接收参数

C.命名关键字形参

def fun(\*,命名关键字形参):

pass

或者

def fun(\*args, 命名关键字形参):

pass

说明，\*前面的形参是0个或多个，\*号后面的形参是1个或多个

作用：约束了\*后所有的参数都必须用关键字传参或者字典关键字传参进行传值

D.双星号字典形参

def fun(\*\*字典形参)

pass

以字典的形式接收参数，通常字典形参命名为kwargs

11、eval()和exec()函数的意义，参数的意义

A.eval(source,global=None,locals=None)

函数的意义：把一个字符串当作表达式来执行，返回表达式执行的结果。

参数的意义：(1)source：必选参数，可以是字符串，也可以是对象，如果是一个字符串，执行字符串的字面量，使用global和locals提供的命名空间，(2)global：提供命名空间，存储全局变量，(3)locals：提供命名空间，存储局部变量。

B.exec(source,global=None,locals=None)

函数的意义：把一个字符串当作Python程序来执行

参数的意义：(1)source：必选参数，表示要执行的Python程序字符串，或者是一个对象，(2)global：同上，(3)locals：同上。

12、什么是闭包，使用的意义

a.将内嵌函数(函数中定义的函数)的语句和环境打包在一起，得到一个对象称为闭包。

b.意义：可以访问函数的局部变量，让这些变量始终保持在内存中，不随外部函数结束而结束。

13、什么是模块

模块是一个包含函数、数据、语句、类等组成的程序组。模块是一个文件，通常以.py结尾。

14、函数的递归计算前N项的和值和阶乘

a.递归计算和值

def sum\_of\_n(n):

if n == 0: # 基本情况：当n为0时，返回0作为和的初始值

return 0

else:

return n + sum\_of\_n(n - 1) # 递归调用自身，计算n + (n-1) + ... + 1

b.递归计算阶乘

def factorial(n):

if n == 0 or n == 1: # 基本情况：0的阶乘和1的阶乘都是1

return 1

else:

return n \* factorial(n - 1) # 递归调用自身，计算n \* (n-1) \* ... \* 1

15、集合的相关操作

集合提供了便捷的方法进行元素的添加、删除和检查，同时也支持对多个集合进行并集、交集和差集等操作，这些功能在数据处理和去除重复元素时非常有用。

16、使用列表推导式定义出符合要求的列表