实验5　函　　数

本实验主要学习用户自定义函数的定义与调用方法、参数的类型、变量的作用域、datetime 标准库的使用方法、递归函数和常见的 Python 内置函数，以及碱基链配对和 Kitty 猫的基因编码两个医药案例的实现。

5.1　实 验 目 的

- 掌握函数的定义方法，包括 def 语句和 lambda 函数、参数的类型以及函数的返回值。
- 掌握函数调用的执行过程。
- 掌握变量的作用域。
- 掌握标准函数库 datetime 中的常用类 datetime、常用属性(datetime.year、datetime.month、datetime.day、datetime.hour、datetime.minute 和 datetime.second)与常用方法(datetime.strftime())等。
- 掌握递归函数。
- 掌握常见的 Python 内置函数。

5.2　知识点解析

5.2.1　函数概述

1. 函数的概念

函数是具有特定书写格式，包含若干条可被作为一个整体执行的语句(函数体)并实现特定功能的语句组。

2. 使用函数编程的优点

函数的作用可以理解成实现某种特定的功能，当人们需要使用这种功能的时候，可以直接调用对应的函数实现。使用函数编程有如下优点：增进代码共享、增加程序易读性、提高代码复用率，以及降低代码维护工作量。

3. Python 函数的类别

Python 的函数可以分为以下 3 类。

① 用户自定义函数：包括定义在主程序文件中的函数和自定义库函数；

② Python 开发者的函数：包括内置函数和标准库函数，内置函数的函数定义集成在 Python 解释器中，标准库函数的定义语句存放在标准库文件中；

③ 第三方库函数：由第三方人员开发，函数定义语句在特定的网络库文件中。

本章主要介绍用户自定义函数的建立和使用方法。

5.2.2 用户自定义函数

1. 定义函数

Python 自定义函数的语法格式如下。

```
def  <函数名> ([参数列表]):
    <函数体语句块>
    [return [返回值列表]]
```

① 函数名可以是任何有效的 Python 标识符；

② 参数列表是调用该函数时传递给它的值，可以有零个、一个或多个；

③ 函数体是函数每次被调用时执行的代码块；

④ 当需要返回值时，使用 return 和返回值列表，否则函数可以没有 return 语句；

⑤ 匿名函数的定义方法：利用 lambda 关键字定义 lambda 函数，它将函数名作为函数结果返回，其语法格式如下。

```
<函数名> = lambda <参数列表>: <表达式>
```

2. 函数的调用

用户自定义函数包括直接定义在主程序文件中的函数和自定义库函数。定义在主程序文件中的函数可以直接调用，自定义库函数则需要和标准库一样 import 后调用。

调用函数的过程分为以下 5 步。

① 主程序在函数调用处暂停；

② 若函数有参数，则将实际参数（简称实参）传递给形式参数（简称形参）；

③ 执行函数体语句；

④ 如果函数有返回值，则将返回值赋给主程序的相应变量或输出；

⑤ 返回主程序，执行调用语句的下一条语句。

3. 函数的参数

参数列表中的参数称为形式参数。形参的类型有 3 种，包括：

① 必选参数；

② 可选参数；

③ 可变数量参数。

其中必选参数在函数调用语句中，必须对应实际参数；可选参数可以不对应实参，调用时如果没有对应实参，函数体语句运行时，可选参数的值取定义语句中的默认值；可变数量参数在不同的函数调用语句中，可以获得可变数量的数据。

4. 参数传递

参数传递时，实参和形参的结合方式分为位置传参和名称传参两种。

① 位置传参：按定义函数时指定的顺序对应传参；

② 名称传参：又称作关键字传参，指通过指定形参名字传入实参值。

5. 函数返回值

return 语句结束函数体语句的运行，并带回返回值。函数体语句中可以出现 0 个或多个 return 语句；函数若不含 return 语句，则该函数无返回值；函数也可以在一条 return 语句

中将 0 个、1 个或同时将多个结果数据赋给调用语句中的主程序的变量。

6. 变量的作用域

作用域是变量的有效范围，由变量的定义位置决定。Python 中的变量按作用域不同，分为全局变量和局部变量。全局变量指在主程序中创建的变量，在整个程序运行期间都有效；局部变量指在函数体语句块中创建的变量，仅在函数被调用执行期间有效，调用结束后被释放。

5.2.3　datetime 库简介

Python 内置的标准库 datetime 中定义了一系列处理日期、时间的函数和类，主要用于实现日期时间类型数据的获取，以及这些数据不同格式的显示。

datetime 库以类的方式提供以下多种表达方式。

① datetime.date：日期表示类，可以表示年、月、日等；

② datetime.time：时间表示类，可以表示小时、分钟、秒、毫秒等；

③ datetime.datetime：日期时间表示的类，功能覆盖 date 类和 time 类；

④ datetime.timedelta：与时间间隔有关的类；

⑤ datetime.tzinfo：与时区有关的信息表示类。

其中最常用的是 datetime 类。datetime 类的常用方法有以下 3 个。

① now()方法：返回一个 datetime 类，表示当前日期和时间；

② utcnow()方法：返回一个 datetime 类，表示当前日期和时间对应的 UTC(世界标准时间)；

③ datetime()方法：构造一个日期和时间类对象。

datetime 类的常用属性如表 5.1 所示。

表 5.1　datetime 类的常用属性

属　　性	描　　述
min	固定返回 datetime 的最小时间对象
max	固定返回 datetime 的最大时间对象
year	返回年份
month	返回月份
day	返回日期
hour	返回小时
minute	返回分钟
second	返回秒钟
microsecond	返回微秒值

datetime 类有以下 3 个常用的时间格式化方法。

① isoformat()方法：采用 ISO 8601 标准显示时间；

② isoweekday()方法：根据日期计算星期后返回 1～7，对应星期一到星期日；

③ strftime(format)方法：根据格式化字符串 format 进行格式显示的方法。

其中 strftime()方法是时间格式化最有效的方法。strftime()方法的格式化控制符如表 5.2 所示。

表 5.2　strftime()方法的格式化控制符

格式化字符串	日期/时间	值范围和实例
%Y	年份	0001～9999，例如：1900
%m	月份	01～12，例如：10
%B	月名	January～December，例如：April
%b	月名缩写	Jan～Dec，例如：Apr
%d	日期	01 ～ 31，例如：25
%A	星期	Monday～Sunday，例如：Wednesday
%a	星期缩写	Mon～Sun，例如：Wed
%H	小时(24h 制)	00 ～ 23，例如：18
%I	小时(12h 制)	00 ～ 11，例如：10
%p	上/下午	AM，PM，例如：PM
%M	分钟	00 ～ 59，例如：26
%S	秒	00 ～ 59，例如：26

5.2.4　递归函数

函数体语句中如果包括调用函数自身的语句，则该函数称为递归函数。递归有两大要素：第一个要素是存在一个或多个基例，基例不需要再次递归，它可以得到确定的结果；第二个要素是函数调用自身。

5.2.5　Python 内置函数

Python 3.8 解释器提供了 69 个可以直接使用的内置函数，如表 5.3 所示。

表 5.3　内置函数列表

abs()	id()	round()	exec()	memoryview()
all()	input()	set()	enumerate()	next()
any()	int()	sorted()	filter()	object()
asci()	len()	str()	format()	property()
bin()	list()	tuple()	frozenset()	repr()
bool()	max()	type()	getattr()	setattr()
chr()	min()	zip()	globals()	slice()
complex()	oct()	bytes()	hasattr()	staticmethod()

续表

dict()	open()	delattr()	help()	sum()
divmod()	ord()	bytearray()	isinstance()	super()
eval()	pow()	callable()	issubclass()	vars()
float()	print()	classmethod()	iter()	__import()__
hash()	range()	compile()	locals()	breakpoint()
hex()	reversed()	dir()	map()	

5.3　实 验 内 容

5.3.1　基本概念题

1. 以下关于函数作用的描述，错误的是(　　)。

A. 复用代码　　B. 提高代码的执行速度

C. 增强代码的可读性　　D. 降低代码的复杂性

2. 下面代码的输出结果是(　　)。

```
def func(a, b):
    a **= b
    return a
s = func(2, 4)
print(s)
```

A. 8　　B. 6　　C. 16　　D. 4

3. 下面代码的输出结果是(　　)。

```
def fun1():
    print('VC', end='*')
def fun():
    for i in range(3):
        fun1()
fun()
```

A. VC*VC*VC*　　B. VC*VC*

C. VC*　　D. ***

4. 以下关于 Python 函数的描述，错误的是(　　)。

A. 使用函数后，代码的维护难度降低了

B. 每次使用函数，需要提供相同的参数作为输入

C. 函数通过函数名进行调用

D. 函数是一段具有特定功能的语句组

5. 下面代码的输出结果是(　　)。

```
def func(a, b):
    return a >> b
s = func(10, 2)
print(s)
```

A. 20　　B. 40　　C. 2　　D. 1

6. 下面代码的输出结果是(　　)。

```
def fib(n):
    a, b = 0, 1
    for i in range(n):
        a, b = b, a+b
    return a
print(fib(9))
```

A. 9　　B. 13　　C. 34　　D. 55

7. Python 中,函数定义可以不包括(　　)。

A. 函数名　　B. 可选参数列表

C. 一对圆括号　　D. 关键字 def

8. 以下代码的描述中,正确的是(　　)。

```
def fact(n):
    s = 1
    for i in range(1, n + 1):
        s *= i
    return s
```

A. 代码中 n 是可选参数　　B. fact(n)函数的功能为求 n+1 的阶乘

C. s 是局部变量　　D. range()函数的范围是[1, n+1]

9. 下面代码的输出结果是(　　)。

```
def func(a):
    a = a + 10
    return a
a = func(10)
b = func(a)
print(a, b)
```

A. 10　20　　B. 10　10　　C. 30　30　　D. 20　30

10. 以下关于 Python 函数参数的描述,正确的是(　　)。

A. 可选参数传递指的是没有传入对应的参数值的时候,就不使用该参数

B. 采用名称传参的时候,实参的顺序需要和形参的顺序一致

C. Python 函数定义中必须对参数指定类型

D. Python 支持按位置传参，也支持按名称传参

11. 下面代码的输出结果是(　　)。

```
def fun(x, y):
    x = y
x = 2
print(fun(x, 4))
```

A. 2　　B. 4　　C. 0　　D. None

12. 下面代码的输出结果是(　　)。

```
def demo(a, b, c, d=5):
    return sum((a, b, c, d))
print(demo(1, 2, 3, 4))
```

A. 10　　B. 11　　C. 6　　D. 出错

13. 下面代码的输出结果是(　　)。

```
def calu(x=3, y=4, z=2):
    return(x * y**z)
h = 4
w = 3
print(calu(h, w))
```

A. 144　　B. 24　　C. 36　　D. 48

14. 下面代码的输出结果是(　　)。

```
def countnum(a, *b):
    print(len(b)+1)
countnum(3, 6, 9, 12, 15, 18)
```

A. 6　　B. 7　　C. 1　　D. 2

15. 关于 Python 的 return 语句，以下选项描述正确的是(　　)。

A. 函数中最多只有一个 return 语句

B. 函数必须有一个 return 语句

C. return 只能返回一个值

D. 函数可以没有 return 语句

16. 下面代码的输出结果是(　　)。

```
t = 5.5
def above_zero(t):
    return t > 0
```

A. 0　　B. False　　C. 5.5　　D. 没有输出

17. 下面代码的输出结果是(　　)。

```
def test(a=20, b=10):
    global z
    z += a * b
    return z
z = 10
print(z, test(), z)
```

A. 10　None　10　　B. 10　210　210

C. UnboundLocalError　　D. 10　200　10

18. 下面代码的输出结果是(　　)。

```
def f():
    print('Python')
print(type(f), type(f()))
```

A. Python
<class 'function'> <class 'function'>

B. Python
<class 'function'> <class 'str'>

C. Python
<class 'function'> <class 'NoneType'>

D. Python
<class 'str'> <class 'function'>

19. 以下关于 Python 函数中变量的描述,错误的是(　　)。

A. 函数里不允许有和函数外同名的变量

B. 全局变量指的是在函数外定义的变量,在程序执行全过程有效

C. 局部变量在函数内部创建和使用,函数调用结束后变量被释放

D. 函数内使用 global 关键字声明简单数据类型变量后,该变量作为全局变量使用

20. 函数表达式 all([1,True,True]) 的结果是(　　)。

A. 无输出　　B. False　　C. 出错　　D. True

21. 给出以下代码,下面选项中描述正确的是(　　)。

```
def fact(n):
    if n == 0:
        return 1
    else:
        return n * fact(n - 1)
num = eval(input("请输入一个数:"))
print(fact(abs(num)))
```

A. 接收用户输入的整数 num，判断 num 是否为素数并输出结论

B. 接收用户输入的整数 num，判断 num 是否为完数并输出结论

C. 接收用户输入的整数 num，输出从 0 到 num 的乘积

D. 接收用户输入的整数 num，输出 num 的绝对值的阶乘

22. 下面代码的输出结果是(　　)。

```
def fun1(a, b, *args):
    print(a)
    print(b)
    print(args)
fun1(1, 2, 3, 4)
```

A. 1
2
[3, 4]

B. 1,2,3,4

C. 1
2
3, 4

D. 1
2
(3, 4)

23. 给出以下代码，下面选项中描述错误的是(　　)。

```
def func(x,y):
    z = x ** 2 + y
    y = x
    return z
x = 10
y = 100
z = func(x,y) + x
```

A. 执行该函数后，变量 z 的值是 200　　B. 该函数的名称为 func

C. 执行该函数后，变量 y 的值是 100　　D. 执行该函数后，变量 x 的值是 10

24. 下面代码的输出结果是(　　)。

```
a = 3
def mya(a, x):
    a = pow(a, x)
    print(a, end=" ")
mya(a, 3)
print(a)
```

A. 3 3　　B. 9 9　　C. 27 3　　D. 27 27

25. 关于 lambda 函数，下面选项中描述错误的是(　　)。

A. lambda 函数也称为匿名函数

B. lambda 函数将函数名作为函数结果返回

C. lambda 的主体是一个表达式

D. lambda 不是 Python 的关键字

26. 下面代码的输出结果是(　　)。

```
MA=lambda x,y:(x<y) * x+(x>y) * y
MI=lambda x,y:(x<y) * y+(x>y) * x
a = 100
b = 200
print(MA(a, b))
print(MI(a, b))
```

A. 200,100　　B. 200,200　　C. 100,100　　D. 100,200

27. 下面代码的输出结果是(　　)。

```
s = lambda x:"yes" if x == 1 else "no"
print(s(0),s(1))
print()
```

A. yes　no　　B. no　yes　　C. 0　yes　　D. no　1

28. 下面代码的输出结果是(　　)。

```
s = ' Amoslin is an antibiotic '
def split(s):
    return s.split('i')
print(s.split())
```

A. [' Amoslin', 'is', 'an', 'antibiotic']　　B. [' Amosl', 'n ', 's an ant', 'b', 'ot', 'c ']

C. 函数定义时报错　　D. 最后一行报错

29. 关于 datetime 库,下面选项中描述错误的是(　　)。

A. datetime 模块包含一个 datetime 类,通过 from datetime import datetime 导入的是 datetime 这个类

B. datetime.now()返回当前日期和时间,其类型是 datetime

C. 用户输入的日期和时间是字符串,要处理日期和时间,必须把 str 转换为 datetime,可以通过 datetime.strftime()实现

D. 特定时间转为 datetime 时间的方式,比如 datetime.datetime(2022,1,1)

30. 关于递归,下面选项中描述错误的是(　　)。

A. 函数定义中调用函数自身的方式称为递归

B. 递归只能存在一个基例

C. 递归的实现通常由分支语句和函数调用构成

D. 递归不是循环

31. 关于 Python 内置函数,下面选项中描述错误的是(　　)。

A. oct()以整数作参数,返回一个以整数值为 Unicode 编码对应的字符

B. max()返回众多参数中的最大值