实验 3　基本数据类型

本实验主要学习 Python 的 3 种基本数据类型的概念及其应用、标准数学库 math 库的用法，以及基于欧几里得距离的化合物相似度计算、清肺排毒汤处方展示、化合物水溶性数据的格式化输出这三个医药数据处理案例的实现。

3.1　实 验 目 的

- 掌握 Python 的 3 种数字类型（整型、浮点型和复数类型）的概念、表示方法、运算操作、内置函数。
- 掌握 Python 字符串类型的概念、表示方法、运算操作、内置函数和方法。
- 掌握 Python 逻辑类型的概念、表示方法、运算操作、应用范围。
- 掌握 Python 不同类型运算操作符的优先顺序。
- 运用 Python 的标准数学库 math 库进行数值计算。
- 掌握基于欧几里得距离的化合物相似度计算方法。
- 掌握清肺排毒汤处方展示、化合物水溶性数据的格式化输出这两个医药数据处理案例的设计与实现，从而进一步熟练掌握实验数据的格式化输出方法。

3.2　知识点解析

3.2.1　Python 的基本数据类型

Python 的数据类型由基本数据类型和组合数据类型组成，其中组合数据类型将会在实验 6 中介绍，而基本数据类型包括以下 3 种。

① 数字类型；

② 字符串类型；

③ 逻辑类型。

3.2.2　数字类型及其操作

1. 数字类型的概念

Python 对数字的表示和使用进行了定义和规范，提供了 3 种数字类型：整型、浮点型和复数类型，分别对应数学中的整数、实数和复数。

2. 数字类型的表示

- 整数类型

Python 整数类型提供了 4 种进制的表示形式。

① 十进制：无引导符，如 1024，－8；

② 二进制：以 0b 或 0B 引导，如 0b110，－0B1001；

③ 八进制：以 0o 或 0O 引导，如 0o137，－0O632；

④ 十六进制：以 0x 或 0X 引导，如 0x5a，－0X1FD。

各进制之间可以通过以下函数相互转换。

① int(x)：返回 x 对应的十进制整数；

② bin(x)：返回 x 对应的二进制整数的字符串；

③ oct(x)：返回 x 对应的八进制整数的字符串；

④ hex(x)：返回 x 对应的十六进制整数的字符串。

• 浮点数类型

Python 中的浮点数有以下两种表示方法。

① 十进制法：如 0.5，－8.，2.85；

② 类科学记数法：1e3，－2.56e－4。

• 复数类型

Python 中的复数与数学中的复数概念一致，即 z＝a＋bj，其中 a 是实部，b 是虚部，虚数单位用 j 或者 J 标识。a 和 b 都是浮点数类型，且 j(或者 J)的前面必须有表示虚部的浮点数 b，即使 b 为 1，也不能省略。可以用 z.real 获得实数部分，用 z.imag 获得虚数部分。

3. 数字类型的运算操作符

Python 内置的数值运算操作符共 9 个，如表 3.1 所示。

表 3.1　Python 内置的数值运算操作符

操　作　符	功　　能
x＋y	计算 x 与 y 之和
x－y	计算 x 与 y 之差
x * y	计算 x 与 y 之积
x / y	计算 x 与 y 之商
x // y	计算 x 与 y 之整数商，即不大于 x 与 y 之商的最大整数
x % y	计算 x 与 y 相除的余数，也称为模运算
－x	x 的负值，即 x *（－1）
＋x	x 本身
x**y	计算 x 的 y 次幂，即 x^y

Python 的所有二元数值运算符都有与之对应的增强赋值操作符，分别为＋＝、－＝、＊＝、/＝、//＝、%＝、**＝，若用符号 op 代替这些操作符，则 x op＝ y 等价于 x ＝ x op y。

4. 内置的数字类型处理函数

Python 内置的数字类型处理函数分以下 3 类。

① 数值运算函数：包括 abs()、divmod()、pow()、round()、max()、min()等。

② 数字类型转换函数：包括 int()、float()、complex()等。

③ 类型判断函数：如 type()。

3.2.3　字符串类型及其操作

1. 字符串类型的概念

Python 对字符串的定义是：由一系列字符组成的序列。字符串字符的序号称为字符的“索引”。Python 中有两种字符串索引编号方式，分别是从左向右的从 0 开始的正向递增索引号和从右向左的从－1 开始的反向递减索引号。

2. 字符串类型的表示

Python 可以使用单引号、双引号或三单引号、三双引号作为字符串的定界符，单引号和双引号可以表示单行字符串，三引号可以表示单行或多行字符串，并且单引号、双引号、三单引号、三双引号还可以互相嵌套，用来表示复杂的字符串。

字符串中有一些特殊字符无法从键盘输入或该字符已经被定义为其他用途，要使用这些字符，必须使用反斜杠“\” 转义这些特殊字符，如“\n”表示换行、“\t”表示制表符(Tab)、“\\”表示反斜杠等。

3. 字符串类型的运算操作符

Python 内置的字符串运算操作符共 5 个，如表 3.2 所示。

表 3.2　Python 内置的字符串运算操作符

操　作　符	功　　能
x＋y	连接两个字符串 x 与 y
x * n 或 n * x	复制 n 次字符串 x
x in s	如果 x 是 s 的子串，则返回 True，否则返回 False
s[i]	索引，返回字符串 s 的第 i 个字符
s[n：m]	切片，返回字符串 s 的索引值从 n 开始到 m－1 的子串(不包含第 m 个字符)

4. 内置的字符串处理函数

Python 内置的字符串处理函数有 6 个，分别是 len()、str()、chr()、ord()、hex()、ocx()。

5. 内置的字符串处理方法

Python 内置的字符串处理方法分以下 6 类。

① 大小写字母转换：包括 s.upper()、s.lower()、s.capitalize()、s.title ()、s.swapcase()等。

② 字符串连接、分割、替换：包括 s.join()、s.split()、s.replace()。

③ 字符串查找和遍历：包括 s.count()、s.find()、s.rfind()、s.index()、s.rindex()等。

④ 字符串判断：包括 s.startswith()、s.endswith()、s.isalnum()、s.isalpha()、s.isdigit()、s.isupper()、s.islower()等。

⑤ 删除字符：包括 s.strip()、s.lstrip()、s.rstrip()等。

⑥ 字符串格式化处理：如 s.format()。

3.2.4　逻辑类型及其操作

1. 逻辑类型的概念

Python 逻辑类型又称为布尔类型。逻辑类型数据只有 True 与 False 两个值。

2. 逻辑类型的表示

逻辑类型数据写作 True 和 False，分别用于表示逻辑真和逻辑假。参与计算时，布尔值也可以当作数值来运算，此时 True 对应的值为整数 1，False 对应的值为整数 0；反之，非 0 可以看作 True，0 可以看作 False。

3. 逻辑类型的运算操作符

Python 内置的常用逻辑运算操作符共 3 个，如表 3.3 所示。

表 3.3　Python 内置的常用逻辑运算操作符

操　作　符	功　　能
and	布尔“与”，如果 x 为 False，则 x and y 返回 False，否则返回 y 的计算值
or	布尔“或”，如果 x 为 True，则 x or y 返回 True，否则返回 y 的计算值
not	布尔“非”，如果 x 为 True，则 not x 返回 False；如果 x 为 False，则 not x 返回 True

4. 返回逻辑类型数据的运算

Python 中，返回逻辑类型数据的情况有以下两类。

① 关系表达式的运算结果；

② 某些判断函数的返回值。

3.2.5　不同类型运算操作符的优先顺序

Python 中，数值运算、字符串运算、关系运算、逻辑运算按优先级递减的顺序排列为：先数值、字符串运算，后关系运算，再逻辑运算。Python 运算符优先级（从高到低）如表 3.4 所示。

表 3.4　Python 运算符优先级（从高到低）

运　算　符	功　　能
**	指数
～、+、−	按位翻转、一元求正数、一元求负数
*、/、//、%	乘、除、求整商、取余
+、−	加法、减法
>>、<<	右移、左移运算符
&	按位与
^、|	按位异或、按位或
<=、<、>、>=	比较运算符
==、!=	等于、不等于运算符
is、is not	身份运算符
in、not in	成员运算符
not、and、or	逻辑运算符
=、%=、/=、//=、−=、+=、*=、**=	赋值与增强赋值运算符

3.2.6　math 库简介

Python 内置的标准库 math 库主要用于提供内置的数学类函数。math 库仅支持整数和浮点数运算，不支持复数运算，一共提供了 4 个数学常数和 44 个函数。4 个数学常数分别为圆周率 math.pi、自然对数 math.e、正无穷大 math.inf 和非浮点数 math.nan。表 3.5 列出了 math 库中的 13 个常用函数。

表 3.5　math 库中的 13 个常用函数

函　　数	功　　能
math.fabs(x)	返回 x 的绝对值
math.fmod(x, y)	返回 x 与 y 的模
math.fsum([x,y,…])	浮点数精确求和
math.ceil(x)	向上取整，返回不小于 x 的最小整数
math.floor(x)	向下取整，返回不大于 x 的最大整数
math.factorial(x)	返回 x 的阶乘，如果 x 是小数或负数，则返回 ValueError
math.gcd(a, b)	返回 a 与 b 的最大公约数
math.pow(x,y)	返回 x 的 y 次幂
math.exp(x)	返回 e 的 x 次幂，e 是自然常数
math.sqrt(x)	返回 x 的平方根
math.log(x[,base])	返回 x 的对数值
math.degrees(x)	角度 x 的弧度值转角度值
math.radians(x)	角度 x 的角度值转弧度值

3.3　实 验 内 容

3.3.1　基本概念题

1. Python 语言提供了 3 种基本的数字类型，它们是(　　)。
   A. 整数类型、浮点数类型、复数类型
   B. 整数类型、二进制类型、浮点数类型
   C. 整数类型、二进制类型、复数类型
   D. 二进制类型、浮点数类型、复数类型
2. 在 Python 语言中，下面是整数的是(　　)。
   A. 2E3　　B. 2e－3　　C. －2000　　D. 2000.0
3. 以下关于二进制整数的定义，正确的是(　　)。
   A. 0B1103　　B. 0b1001　　C. 0B102　　D. 0B10A

4. 以下不是 Python 的数值运算操作符的是(　　)。

A. $　　B. /　　C. **　　D. //

5. 表达式 5.0－8 * 2 ** 3 % 5 // 3－True 的值是(　　)。

A. 报错　　B. 3　　C. 3.0　　D. 4.0

6. int (3＋0j)的返回值是(　　)。

A. 3　　B. 3.0　　C. 0.0　　D. TypeError

7. divmod(10,3)的返回值是(　　)。

A. [3,1.0]　　B. (3,1)　　C. [3,1]　　D. (3.0,1)

8. 以下代码的输出结果是(　　)。

```
x, y = 4.0, 4
y -= x
print(y)
```

A. 0　　B. 0.0　　C. '0'　　D. '0.0'

9. 以下代码的输出结果是(　　)。

```
print(0.1 + 0.2 == 0.3)
```

A. －1　　B. True　　C. False　　D. 0

10. 以下代码的输出结果是(　　)。

```
a = 3.14
print(complex(a))
```

A. 0.14　　B. 3.14i＋j　　C. 3.14　　D. (3.14＋0j)

11. 以下关于 Python 的描述中,正确的是(　　)。

A. Python 的整数类型有长度限制,一旦超过上限,会产生溢出错误

B. Python 语言中采用严格的"缩进"表明程序格式,不可嵌套

C. Python 中可以用八进制表示整数

D. Python 的浮点类型没有长度限制,只受限于内存的大小

12. 以下关于 Python 语言复数类型的描述中,错误的是(　　)。

A. 复数可以进行四则运算

B. 实部不可以为零

C. Python 语言中可以使用 z.real 和 z.imag 分别获取 z 的实部和虚部

D. 复数类型与数学中复数的概念一致

13. 已知复数 z 的值为 1.23e－4＋5.6e＋78j,则 z.real 的值为(　　)。

A. 1.23　　B. 1.23e－4＋5.6e　　C. 5.6e＋78　　D. 0.000123

14. 以下关于 Python 字符串的描述中,错误的是(　　)。

A. 在 Python 字符串中,可以混合使用正整数和负整数进行索引和切片

B. Python 字符串采用[N:M]格式进行切片,获取字符串从索引 N 到 M 的子字符串(包含 N 和 M)

C. 字符串中的第一个“\”表示转义符

D. 空字符串可以表示为""或''

15. 以下关于 Python 字符串的描述中，正确的是(　　)。

A. 字符应视为长度为 1 或为 2 的字符串

B. 字符串中的字符可进行数学运算，但进行数学运算的字符必须为数字

C. 在三引号字符串中可包含换行、回车等特殊的字符

D. 字符串可以进行切片赋值

16. 以下关于 Python 字符编码的描述中，正确的是(　　)。

A. Python 字符编码使用 ASCII 编码存储

B. chr(x)和 ord(x)函数用于在单字符和 Unicode 编码值之间进行转换

C. print(chr('a'))输出 97

D. print(ord(65))输出 A

17. Python 语言中，以下表达式结果为 False 的选项是(　　)。

A. "CD" < "CDEF"　　B. "DCBA" < "DC"

C. "" < "G"　　D. "love" > "LOVE"

18. 以下代码的输出结果是(　　)。

```
s = 'C\tP\bR'
print(len(s))
```

A. 3　　B. 5　　C. 7　　D. 6

19. 以下代码的输出结果是(　　)。

```
a,b,c,d = 'Drug'
print(a + c)
```

A. Dg　　B. Du　　C. DrugDrug　　D. TypeError

20. 以下可以在屏幕上输出 C:\tablet 的是(　　)。

A. print("C:\tablet")　　B. print("C: '\ tablet'")

C. print("C:\\ tablet ")　　D. print("C: '\'tablet ")

21. 以下表达式的结果是"年少好友"的是(　　)。

A. "友好少年"[0::]　　B. "友好少年"[0: -1: 1]

C. "友好少年" [::-1]　　D. "友好少年" [0::-1]

22. 字符串 s = 'dolantin'，显示结果为“lan”的选项是(　　)。

A. print(s[3: 6])　　B. print(s[2: 4])

C. print(s[-6: 5])　　D. print(s[2: -4])

23. 字符串 s = 'Penicillin'，则表达式 s[1::3]的值为(　　)。

A. 'ecl'　　B. 'Piln'　　C. 'eiiln'　　D. 'en'

24. 以下返回值为“世界这么大，我想去溜达！”的语句是(　　)。

A. "哈哈哈哈世界这么大哈，我想去溜达！ 哈哈哈哈哈".strip("哈")

B. "哈哈哈哈世界这么大哈，我想去溜达！ 哈哈哈哈哈".replace("哈","")

C. "哈哈哈哈世界这么大哈,我想去溜达! 哈哈哈哈哈".find("哈")

D. "哈哈哈哈世界这么大哈,我想去溜达! 哈哈哈哈哈".zfill(12)

25. 以下有关 Python 字符串类型的操作描述,正确的是(　　)。

A. 想把一个字符串 str 所有的字符都大写,用 upper(str)

B. 设 x = 'aaa',则执行 x / 3 的结果是 'a'

C. 想获取字符串 str 的长度,用字符串处理函数 len(str)

D. str.isnumeric()方法把字符串 str 中的数字字符串变成数字

26. ",".join([1, 2, 3, 4, 5])的返回值为(　　)。

A. 12345　　B. "1,2,3,4,5"　　C. TypeError　　D. 15

27. 以下代码的输出结果是(　　)

```
s = "GS,NS,NG,NE."
print(s.split())
```

A. ['GS', 'NS', 'NG', 'NE.']　　B. ['GS', 'NS', 'NG', 'NE']

C. ['GS,NS,NG,NE.']　　D. ['GS,NS,NG,NE']

28. 以下代码的输出结果是(　　)

```
a = 'CPR'
b = 'AED'
print("{:*>10}:{:*<10}".format(a,b))
```

A. *******CPR: AED*******　　B. CPR*******: *******AED

C. CPR: AED　　D. *******CPR: *******AED

29. 以下代码的输出结果是(　　)

```
x = 3.1415926
print("{} {:6}".format(round(x), round(x, 3)))
```

A. 3.0 3.1415　　B. 3　3.1415

C. 3　3.142　　D. 3.0　3.142

30. 以下代码段执行后,用户输入 10,程序的运行结果为(　　)。

```
x = eval(input("please input a number:"))
if 20:
    print(True == 1)
else:
    print('False == 0')
```

A. True == 1　　B. True

C. False == 0　　D. False

3.3.2　简单操作题

1. 把以下算术表达式改写为 Python 表达式。

(1) b^2+4ac

(2) $\frac{3}{4}(|x-6|+y)^2$

(3) $\frac{(x+y)^2}{5x+\sqrt{y}}$

(4) $\frac{13\bmod 5+2^3}{|a+b|}$

2. 请先自行计算出下列表达式的值，再在 IDLE Shell 中输入并验证结果。

(1) 11 % -3 +2 * 3 ** 2 + 18 // 4 - True

(2) (4 / 2 + 5j) * 3j

(3) (ord('Z')-ord('A') + 3) % 26

(4) (len('Aspirin') + len('Amoxil')) / 2

3. 请先自行估算出下列表达式的值，再在 IDLE Shell 中调用 math 库的函数，计算并验证下列表达式的值。

(1) math.sin(math.pi/2)

(2) math.sqrt(math.pow(3,2))

(3) math.fabs(math.floor(-4.6))

(4) math.gcd(243,18)

4. 请先自行计算多项式 ax^3-bx^2+cx-d 的值，其中 a、b、c、d 的值分别为 1、2、3、4，x 的值为 3.14，再在 IDLE Shell 中验证计算结果。

5. 已知字符串 s1 = 'Aspirin'，s2 = 'Amoxil'，s = s1+s2，试计算以下表达式的值：s[0]、s[-1]、s[2：5]、s[::-1]、s[::3]、s[2：8：2]、s.upper()、s.find('A')、s.find('A',1)，且在 IDLE Shell 中输入并验证结果。

6. 计算直角三角形的面积。

要求：从键盘输入直角三角形的两条直角边长度，计算直角三角形的面积并在屏幕上输出结果，程序运行效果如下。

```
输入三角形第一条直角边长:3
输入三角形第二条直角边长:4
直角三角形的面积为:6.0
```

分析：两条直角边的输入可通过 eval()函数嵌套 input()函数实现，三角形面积的计算可以通过编写直角三角形面积公式的 Python 表达式实现，结果输出可以通过 print()函数实现。

实验步骤：

1) 添加并完善程序代码

新建文件，输入以下代码，填写正确代码以替换横线，不修改其他代码，实现题目功能。

```
a = eval(input("输入三角形第一条直角边长:"))
b = eval(input("输入三角形第二条直角边长:"))
area =_______
print("直角三角形的面积为:{:.1f}".format(_______))
```

2）保存并运行程序

将文件保存为 PY30206.py，运行程序，验证程序的正确性并观察程序执行效果。

3）提示

请注意直角三角形通过两条直角边长计算面积公式转换成 Python 语句的正确表达方式。

7. 实现基本统计量计算。

要求：假设一组数据表示为 $S=s_0,s_1,\cdots,s_{n-1}$，请编程实现其最大值、最小值、平均值、标准差等基本统计量计算。其中算术平均值如式(3.1)所示，标准差如式(3.2)所示。

$$\text{算术平均值：mean}=\left(\sum_{i=0}^{n-1}s_i\right)/n \tag{3.1}$$

$$\text{标准差：dev}=\sqrt{\left(\sum_{i=0}^{n-1}(s_i-\text{mean})^2\right)/(n-1)} \tag{3.2}$$

程序运行效果如下。

```
请输入用逗号分隔的 5 个数:1,2,3,4,5
最大值为: 5
最小值为: 1
平均值为: 3.0
标准差为: 1.5811388300841898
```

分析：可通过逗号表达式及 eval()函数嵌套 input()函数实现基本统计数据的输入，最大值和最小值可通过 Python 解释器内置数值运算函数 max()和 min()实现，平均值和标准差可通过将数学公式转换成 Python 表达式获得。

实验步骤：

1）添加并完善程序代码

新建文件，输入以下代码，填写正确代码以替换横线，不修改其他代码，实现题目功能。

```
a, b, c, d, e = eval(input("请输入用逗号分隔的 5 个数:"))
sum =_____________________
mean =_____________________
dev =_____________________
print("最大值为:", max(a, b, c, d, e))
print("最小值为:", min(a, b, c, d, e))
print("平均值为:", mean)
print("标准差为:", dev)
```

2）保存并运行程序

将文件保存为 PY30207.py，运行程序，验证程序的正确性并观察程序执行效果。

3）提示

本题处理了包含 5 个元素的基本统计量计算，分别使用变量 a、b、c、d、e 存储和处理基本数据，请思考并查阅资料，若输入前并不知道一共有多少个数据需要输入和统计，这样的需求在 Python 中该如何实现？

8. 通过月份值提取月份英文简写。

要求：从键盘输入一个月份数字，返回对应月份名称简写，程序运行效果如下。

```
请输入月份(1~12) :8
月份简写是: Aug
```

分析：本题中将所有的月份英文简写排列成一个连续的字符串，程序执行时接收用户输入的 1～12 中的某个月份，由于每个月份的英文简写都由 3 个字符构成，所以可通过给定的月份由公式计算出目标月份英文简写在已知字符串中的位置，再进行字符串切片处理，提取出目标月份英文简写子串并输出。

实验步骤：

1）添加并完善程序代码

新建文件，输入以下代码，填写正确代码以替换横线，不修改其他代码，实现题目功能。

```
n = input("请输入月份(1~12) :")
months = "JanFebMarAprMayJunJulAugSepOctNovDec"
pos = (int(n) - 1) * 3
monthAbb = months[_____]
print(_____)
```

2）保存并运行程序

将文件保存为 PY30208.py，运行程序，验证程序的正确性并观察程序执行效果。

3）提示

需要熟练掌握字符串单个字符索引和字符串切片规则，其中字符串切片可以通过两个索引值确定一个位置范围，返回这个范围的子串，格式为：＜string＞[＜start＞：＜end＞]，其中 start 和 end 都是整数型数值，这个子序列从索引 start 开始，直到索引 end 结束，但不包括 end 位置。

9. 按要求格式输出给定字符串。

要求：接收从键盘输入的字符串，按题目要求把该字符串内容输出到屏幕上，具体格式要求为：输出宽度为 30 个字符，不足部分以星号“*”填充，居中对齐。如果输入的字符串长度超过 30，则全部输出，程序运行效果如下。

```
请输入一个字符串:Aspirin
************Aspirin*************
```

分析：Python 中采用内置的字符串处理方法 format()实现字符串输出格式控制。

实验步骤：

1）添加并完善程序代码

新建文件，输入以下代码，填写正确代码以替换横线，不修改其他代码，实现题目功能。

```
s = input("请输入一个字符串:")
print("{_____}".format(s))
```

2）保存并运行程序

将文件保存为 PY30209.py，运行程序，验证程序的正确性并观察程序执行效果。

3）提示

Python 内置的字符串输出控制方法 format()的调用格式为：＜模板字符串＞.format(＜以逗号分隔的参数＞)，其中，＜模板字符串＞ 包括需原样返回的字符和槽，槽规定如何格式化参数得到子串，它除包括参数索引号，还包括格式控制信息。此时，槽的内部样式如下：{＜参数索引号＞：＜格式控制标记＞}，其中格式控制标记用来控制参数显示时的格式。格式控制标记包括＜填充＞、＜对齐＞、＜宽度＞、＜千位分隔符＞、＜.精度＞、＜类型＞ 6 个字段，这些字段都是可选的，可以组合使用。

3.3.3 综合应用题

1. 实现给定十进制整数的多进制转换

要求：编写程序，实现将从键盘输入的十进制整数转换为二进制、八进制和十六进制(大写)形式并按要求输出到屏幕上，各进制数之间采用制表符“\t”分隔，程序运行效果如下。

```
输入数字:350
对应的二进制数:101011110    八进制数:536    十六进制数:15E
```

分析：Python 中的十进制向 n 进制的转换可以采用除 n 取余的常规算法实现。本题针对的是特殊进制，如二进制、八进制、十六进制，则可以简单地采用 format ()方法设置对象输出显示时格式控制标记中的类型字段实现，其中 b 表示二进制方式、o 表示八进制方式、X 表示大写十六进制方式。

实验步骤：

1）添加并完善程序代码

新建文件，输入以下代码，请在……处使用一行或多行代码替换，不修改其他代码，实现题目功能。

```
num = eval(input("输入数字:"))
……
```

2）保存并运行程序

将文件保存为 PY30301.py，运行程序，验证程序的正确性并观察程序执行效果。

3）提示

Python 中还可以采用内置的进制转换函数完成进制转换的工作。主要的进制转换函数如下。

int(x)：返回 x 对应的十进制整数。

bin(x)：返回 x 对应的二进制整数的字符串。

oct(x)：返回 x 对应的八进制整数的字符串。

hex(x)：返回 x 对应的十六进制整数的字符串。

例如：int(0o123)的返回值为 83；bin(3)的返回值为'0b11'。转换得到的二进制、八进

制、十六进制字符串会带有进制前缀。

2. 实现 3 位正整数的分解

要求：编写程序，实现分别提取出从键盘输入的 3 位正整数的百、十和个位上的数字，并按要求将它们输出到屏幕上，程序运行效果如下。

```
请输入任意一个 3 位正整数:519
519 的百位是:5,十位是:1,个位是:9
```

分析：提取 3 位正整数，每位上的数字可以通过 Python 提供的内置算术运算整除(/)、取余(%)或者两者的配合来实现。

实验步骤：

1）添加并完善程序代码

新建文件，输入以下代码，请在……处使用一行或多行代码替换，不修改其他代码，实现题目功能。

```
n = eval(input("请输入任意一个 3 位正整数:"))
……
print("{}的百位是:{},十位是:{},个位是:{}".format (n,b,s,g))
```

2）保存并运行程序

将文件保存为 PY30302.py，运行程序，验证程序的正确性并观察程序执行效果。

3）提示

本题还有另外一种解决思路，即将该正整数转换为字符串，利用对字符串的单个字符的索引访问获得不同位上的数字，请尝试实现该方法。

3. 实现将给定字符串中每个单词的首字母转换为大写

要求：用户从键盘输入一行字符，编写程序，实现将该行字符中每个单词的首字母都转换为大写并输出到屏幕上，程序运行效果如下。

```
请输入一行西文字符:mankind will be able to overcome the virus epidemic.
转换后的内容为: Mankind Will Be Able To Overcome The Virus Epidemic.
```

分析：Python 提供了将西文文本中每个单词首字母转换为大写的内置字符串处理方法 title()。

实验步骤：

1）添加并完善程序代码

新建文件，输入以下代码，请在……处使用一行或多行代码替换，不修改其他代码，实现题目功能。

```
s = input("请输入一行西文字符:")
……
```

2）保存并运行程序

将文件保存为 PY30303.py，运行程序，验证程序的正确性并观察程序执行效果。

3）提示

本题还可以不通过调用内置函数方法，而是自主编程，实现将西文字符串中的每个单词首字母转换为大写。首先借助 split()方法进行字符串分解，然后逐个转换每个单词的首字母为大写，再借助 join()方法将转换后的单词连接成新的字符串。可在学习循环结构后尝试实现该方法。

4. 实现指定 Unicode 码值到字符的转换并按要求格式输出

要求：从键盘输入一个 9800～9811 的正整数 n，把 n－1、n、n＋1 这 3 个数值对应的 Unicode 编码字符按照给定格式要求输出到屏幕上，具体格式要求为：输出宽度为 15 个字符，不足部分用“＃”填充，居中对齐，程序运行效果如下。

```
请输入一个 9800~9811 的数字:9805
######♌ ♍ Ω######
```

分析：Python 中采用内置的字符串处理函数 chr()实现由 Unicode 码值向对应字符的转换，另外，采用内置的字符串处理方法 format()实现格式输出控制。

实验步骤：

1）添加并完善程序代码

新建文件，输入以下代码，请在……处使用一行或多行代码替换，不修改其他代码，实现题目功能。

```
n = eval(input("请输入一个 9800~9811 的数字:"))
……
```

2）保存并运行程序

将文件保存为 PY30304.py，运行程序，验证程序的正确性并观察程序执行效果。

3）提示

Python 3.x 以 Unicode 字符为计数基础，因此英文字符和中文字符都是 1 个长度单位。Python 内置的 chr(x)函数用来返回 Unicode 码值 x 对应的单字符；ord(s)函数用来返回单字符 s 对应的 Unicode 码值，它们是一对反函数。

5. 实现单字符的凯撒密码加密

要求：凯撒密码是古罗马凯撒大帝用来对军事情报进行加密的算法，它是一种替换加密的技术。本题采用的替换规则是将信息中的每个英文字符替换为字母表序列该字符后面第 3 个字符，即替换偏移量为 3，对应关系如图 3.1 所示。

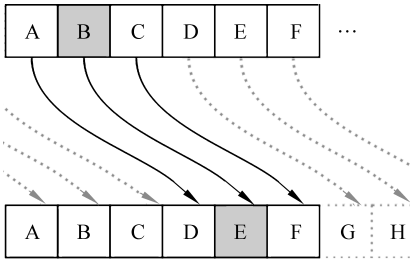

图 3.1　凯撒加密规则

请编写程序，完成单个字符的凯撒加密，程序运行效果如下。

```
请输入一个大写字母(A~Z):X
凯撒加密后的字符为: A
```

分析：首先需要进一步明确本题中单个字符凯撒加密的规则：若输入的单个字符非大写字母，则不做加密处理，密文和明文一致；若输入的是一个大写字母，则对明文字母进行字母表中向后推 3 位的处理，获得密文字母，若向后推时已超出字母表范围，则转回头从 A 开始继续计数推移。这里，字母表中的位置推移需要用到 Python 提供的字符内置函数 chr() 和 ord()。可以先用 ord()函数将明文字符转换成对应的 Unicode 码值，将码值加 3 后，再利用 chr()函数将其转换回密文字符。

实验步骤：

1）添加并完善程序代码

新建文件，输入以下代码，请在……处使用一行或多行代码替换，不修改其他代码，实现题目功能。

```
old = input("请输入一个大写字母(A~Z):")
……
print("凯撒加密后的字符为:",new)
```

2）保存并运行程序

将文件保存为 PY30305.py，运行程序，验证程序的正确性并观察程序执行效果。

3）提示

在进行加密处理的过程中，可以分推移后未超出字母表和超出字母表两种情况讨论，也可以将取模运算“%”和字母表中的字母个数为 26 等已知因素合并成统一的推导公式进行处理，请思考两种处理方式并尝试实现。

3.3.4　医药案例题

1. 基于欧几里得距离的化合物相似度计算

要求：现假设衡量化合物相似度的依据是水溶性和血红蛋白结合率。定义化合物 A 和 B 之间的相似度，如式(3.3)所示。

$$s = \sqrt{(F1_A - F1_B)^2 + (F2_A - F2_B)^2} \tag{3.3}$$

式中，F1 表示水溶性，F2 表示血红蛋白结合率，请编程实现：输入两个化合物 A、B 的水溶性和血红蛋白结合率数值，计算它们之间的相似度并输出，结果保留 2 位小数。程序运行效果如下。

```
请输入 A 的水溶性数值:32.2
请输入 A 的血红蛋白结合率:21
请输入 B 的水溶性数值:28.6
请输入 B 的血红蛋白结合率:23.2
所求相似度为:4.22
```

分析：以 A 的水溶性数值为例，它在程序中的使用分两步：先从键盘输入，然后用于相似度计算。因此，本例首先需要创建 4 个变量，分别用于存放用户从键盘输入的两种化合物的水溶性数值及血红蛋白结合率值，再列出对应计算公式的 Python 表达式计算相似度，按要求输出即可。

实验步骤：

1）添加并完善程序代码

新建文件，输入以下代码，填写正确代码以替换横线，不修改其他代码，实现题目功能。

```
f1a = eval(input("请输入 A 的水溶性数值:"))
f2a =_____________
f1b =_____________
f2b =_____________
s =_____________
print("所求相似度为:{:.2f}".format(s))
```

2）保存并运行程序

将文件保存为 PY30401.py，运行程序，验证程序的正确性并观察程序执行效果。

3）提示

在数据分析和数据挖掘以及搜索引擎中，经常需要知道个体间差异的大小，进而评价个体的相似性和类别。相似度就是比较两个事物的相似性。一般通过计算事物特征之间的距离实现，如果距离小，那么相似度大；如果距离大，那么相似度小。常用的相似度计算公式有欧几里得距离、曼哈顿距离、明科夫斯基距离、余弦相似度等。化合物相似度在化学信息学和药物发现中具有悠久的历史，许多计算方法采用相似度测定鉴定研究的新化合物。第 2 章案例中实现了通过对两种化合物的分子指纹计算其余弦相似度，本题采用欧几里得距离法实现化合物水溶性和血红蛋白结合率相似度计算。大家可依据具体的要求和场景选择更多的相似度计算方法进行研究。

2. 清肺排毒汤处方的格式化展示

要求：请将存放于一个字符串中的中药清肺排毒汤处方按指定要求展示，原始处方字符串为

```
formula="麻黄 9g、炙甘草 6g、杏仁 9g、生石膏 15~30g(先煎)、\
桂枝 9g、泽泻 9g、猪苓 9g、白术 9g、茯苓 15g、柴胡 16g、黄芩 6g、\
姜半夏 9g、生姜 9g、紫菀 9g、冬花 9g、射干 9g、细辛 6g、山药 12g、\
枳实 6g、陈皮 6g、藿香 9g"
```

新的输出显示要求为：标题“清肺排毒汤”居中显示；正文每行显示 4 味中草药，每味中草药及其用量信息占 10 个字符宽度；行间距是一行（输出一个空行）。程序运行效果如下。

```
                          清肺排毒汤

麻黄 9g        炙甘草 6g      杏仁 9g        生石膏 15~30g(先煎)

桂枝 9g        泽泻 9g        猪苓 9g        白术 9g

茯苓 15g       柴胡 16g       黄芩 6g        姜半夏 9g
```

```
生姜 9g          紫菀 9g          冬花 9g          射干 9g
细辛 6g          山药 12g         枳实 6g          陈皮 6g
藿香 9g
```

分析：本题原始处方字符串中各味中草药之间是用“、”分隔的，而新的输出显示中各味中草药各占宽度相等的输出位置，并且没用分隔符进行切分，因此需要对原始字符串进行切分，分离出各味中草药，进行相同宽度的左对齐输出，且控制每行输出 4 味中草药后换行，另输出一个空行作为行与行之间的间隔。由于本题需要提前用到循环结构控制语句，因此核心代码后都给出了注释说明供参考。

实验步骤：

1）添加并完善程序代码

新建文件，输入以下代码，填写正确代码以替换横线，不修改其他代码，实现题目功能。

```
formulaName = "清肺排毒汤"
formula = "麻黄 9g、炙甘草 6g、杏仁 9g、生石膏 15~30g(先煎)、\
桂枝 9g、泽泻 9g、猪苓 9g、白术 9g、茯苓 15g、柴胡 16g、黄芩 6g、\
姜半夏 9g、生姜 9g、紫菀 9g、冬花 9g、射干 9g、细辛 6g、山药 12g、\
枳实 6g、陈皮 6g、藿香 9g"

print(formulaName.center(40),"\n")  #处方名称居中输出,并输出一个空行
n = formula.count("、")             #计算顿号个数->n
remain = formula                    #给 remain 赋初值 formula
for i in range(1,n + 1):            #实现对前 n 味中草药采用统一方法处理
   pos = _____________              #查找"、"的位置->pos
   herb = _____________             #以 pos 为界,切片得到第一味中草药的字符串
   herb = "{:<10}".format(herb)     #用 format()方法格式化 herb
   print(herb, end="")              #输出格式化后的 herb 的信息
   if i % 4 == 0:                   #控制每行输出 4 味中草药,并加入空行
       print("\n")
   remain = _____________           #更新 remain,将处理过的中草药从 remain 中去掉
print(remain)                       #输出最后一味中草药的信息
```

2）保存并运行程序

将文件保存为 PY30402.py，运行程序，验证程序的正确性并观察程序执行效果。

3）提示

对存在固定分隔符的字符串进行字符串分解的问题，可以采用本题类似的字符串逐个查找分隔符进行切片分解的方法，还可以通过字符串内置的处理方法 split()进行根据指定分隔符的字符串切分操作，所得结果为由切分出的子串组成的列表。列表作为最重要的组合数据类型将在第 6 章实验中详细学习和使用。

3. 化合物水溶性数据的格式化输出

要求：实验测得的若干化合物的水溶性数据记录在一个长字符串 s 中：s = "IdentityNo.,Solubility\nLT-615-348,0.019714\nLT-771-215,0.03072346"，由于 s 中包含了换行符“\n”，因此若直接用 print 语句对 s 进行输出，则显示结果如下。

```
IdentityNo.,Solubility
LT-615-348,0.019714
LT-771-215,0.03072346
```

编程实现以下格式化输出，新的输出显示要求为：用中文“化合物编号”和“水溶性值”替代原字符串中的英文标签，解析出的化合物编号靠左对齐，而水溶性值靠右对齐，程序运行效果如下。

```
化合物编号      水溶性值
LT-615-348       0.0197
LT-771-215       0.0307
```

分析：本题原始数据字符串中是用换行符“\n”分隔一行标签和两行化合物编号、水溶性值数据，因此需要借助换行符“\n”所在的位置进行字符串的第一层分解，而每行数据之间又是通过逗号“,”进行分隔的，因此需要对第一层分解出的每行语句再根据“,”进行第二层分解，将两次分解后的结果按要求格式进行输出控制，具体分解方法可参考第 2 题。

实验步骤：

1）添加并完善程序代码

新建文件，输入以下代码，请在……处使用一行或多行代码替换，不修改其他代码，实现题目功能。

```
s = "IdentityNo.,Solubility\nLT-615-348,0.019714\nLT-771-215,0.03072346"
print(s)
……
print("化合物编号{:>8}".format("水溶性值"))
print("{:<10}{:>12.4f}".format(id1, sValue1))
print("{:<10}{:>12.4f}".format(id2, sValue2))
```

2）保存并运行程序

将文件保存为 PY30403.py，运行程序，验证程序的正确性并观察程序执行效果。

3）提示

请尝试采用两种方法实现以上功能：一是采用字符串内置方法 find()查找分隔符位置，然后进行字符串切片；二是通过字符串内置方法 split()将已知分隔符的字符串分解为列表，然后对列表元素进行下一步的分解处理。

3.4 实验解析

3.4.1 基本概念题解析

1. 答案：A

解析：Python 的数字类型包括整数类型、浮点数类型、复数类型 3 种，二进制是整数类型的一种表示形式，故本题选择 A 选项。

2. 答案：C

解析：Python 中的浮点数类型拥有十进制法和包含 e 的类科学记数法两种表示形式，因此，2E3、2e－3 和 2000.0 均为浮点数，故本题选择 C 选项。

3. 答案：B

解析：Python 提供的整数类型有 4 种进制的表示形式，分别是二进制、八进制、十进制和十六进制。其中，二进制以 0b 或 0B 引导，且二进制的数码只有 0 和 1，进位规则是逢二进一，故本题选择 B 选项。

4. 答案：A

解析：Python 内置的数值运算操作符有 9 个，分别是＋、－、＊、/、//、％、－(单目)、＋(单目)、＊＊，故本题选择 A 选项。

5. 答案：C

解析：Python 算术表达式的计算优先级与相应的数学表达式的计算优先级相同，本题中的表达式 5.0－8 ＊ 2 ＊＊ 3 ％ 5 // 3－True 的计算顺序为，首先计算 2 ＊＊ 3 结果为 8，然后计算 8 ＊ 8 结果为 64，接着计算 64 ％ 5 结果为 4，计算 4 // 3 结果为 1，计算 5.0－1 结果为 4.0，最后计算 4.0－True，由于此时 True 取值为 1，因此最终结果为 3.0。计算过程中需注意 Python 中不同数字类型之间可以进行混合运算，运算后生成的结果为最宽类型，故本题选择 C 选项。

6. 答案：D

解析：Python 内置的 int() 函数用于将一个数值或数值型字符串转换为十进制的整数形式，前提是转换对象能够成功取整或者转换为一个整数。由于 3＋0j 是一个复数，Python 中规定一个复数即使虚部为 0，也不能转换成一个整数，故本题选择 D 选项。

7. 答案：B

解析：Python 内置的 divmod()函数把两个参数的整除和取余运算结果结合起来，返回一个包含整除商和余数的元组，故本题选择 B 选项。

8. 答案：B

解析：Python 语法中允许通过采用逗号表达式的方式一次性给多个变量赋初值，故 x，y ＝ 4.0，4 同时完成了 x ＝ 4.0，y ＝ 4 的赋值，y －＝ x 中的“－＝”是 Python 的增强赋值操作符，相当于执行语句 y ＝ y － x，由于 y 的值是 4，而 x 的值是 4.0，一个整数与一个浮点数相减时，Python 的计算规则是不同数字类型之间可以进行混合运算，运算后生成的结果为最宽类型，即整数和浮点数混合运算，输出结果是浮点数，因此该语句执行后 y 中的值为浮点数 0.0，故本题选择 B 选项。

9. 答案：C

解析：Python 中的“＝＝”是一个关系运算符，功能是判断左右两边表达式的值是否相等，本题中的表达式 0.1＋0.2 ＝＝ 0.3 在数学含义上判断是否相等的结果应是 True，而真正运行时会发现输出的结果是 False，这是因为整数或浮点数在用计算机进行存储和运算时都会转换成二进制的形式，同十进制一样，二进制也会存在无限循环小数，而计算机对浮点数的表示通常是有二进制位数限制的，这使得在二进制的表达上必须采用取近似值的手段丢掉一些二进制位，因此会造成不可避免的浮点数精度丢失问题，所以才会出现本题中的特例情况，故本题选择 C 选项。

10. 答案：D

解析：Python 内置的 complex() 函数用于创建一个复数或者将一个数值或数值型字符串转换为复数形式，其返回值为一个复数，且虚部为 0 时，也不可以被省略，故本题选择 D 选项。

11. 答案：C

解析：Python 语言中采用严格的“缩进”表明程序格式，结构之间是可以产生嵌套的；Python 的整数类型理论上没有取值范围的限制，实际上的取值范围仅受限于运行 Python 程序的机器的内存大小，而浮点数类型与整数类型由计算机的不同硬件单元执行，处理方法也不同，Python 浮点数的取值范围和小数精度受不同计算机系统的限制，超出上限时会产生溢出错误；Python 中为整数类型提供了二进制、八进制、十进制和十六进制 4 种进制表示，故本题选择 C 选项。

12. 答案：B

解析：Python 中的复数与数学中的复数概念一致；z = a+bj，其中 a 是实部，b 是虚部，虚数单位用 j 或者 J 标识。a 和 b 都是浮点数类型，若 a 为 0，则可以省略实部，但必须有表示虚部的浮点数 b 和 j(或者 J)，即使 b 为 0，也不能省略 j，b 为 1，也不能仅记为 j，而需记为 1j；复数可以进行四则运算；可以用 z.real 获得实数部分，用 z.imag 获得虚数部分，故本题选择 B 选项。

13. 答案：D

解析：对于本题中的复数 z，其实部为 1.23e-4，虚部为 5.6e+78j，因此 z.real 的值应为 1.23e-4，显示为 0.000123，故本题选择 D 选项。

14. 答案：B

解析：Python 可以使用单引号、双引号或三单引号、三双引号作为字符串的定界符，因此空字符串可以表示为""或'';字符串中有一些特殊字符无法从键盘输入或该字符已经被定义为其他用途，要使用这些字符，就必须使用反斜杠“\”引导转义这些特殊字符；Python 允许混合使用正整数和负整数进行索引和切片；Python 字符串采用[N:M]格式进行切片，获取字符串从索引第 N 到第 M 的子字符串(包含 N，但不包含 M)，故本题选择 B 选项。

15. 答案：C

解析：Python 中的字符串没有固定长度的限制；Python 字符串除了可以由“+”完成两个字符串的连接，由“*”完成重复多次连接一个字符串生成新的字符串外，不能进行数学运算，即使字符串中的字符都为数字也不可以；三引号可以表示单行或多行字符串，因此，在三引号字符串中可包含换行、回车等特殊的字符；由于字符串属于固定类型，因此不可以进行切片赋值操作，故本题选择 C 选项。

16. 答案：B

解析：Python 字符使用 Unicode 编码表示和存储；chr(x)和 ord(x)函数是一对反函数，用于在单字符和 Unicode 编码值之间进行转换；chr(x)返回 Unicode 编码 x 对应的单字符，而 ord(x)用于返回单字符对应的 Unicode 编码，print(chr('a'))和 print(ord(65))语句中两个函数的实参类型均出错，故本题选择 B 选项。

17. 答案：B

解析：Python 字符串可以进行大小比较，其比较规则为，按索引号由小到大的顺序，把