

第 3 章

练习题参考答案

3.1 填空题参考答案

1. 解释
2. 交互式、文件式
3. 面向对象
4. py
5. import
6. PyInstaller
7. dist
8. pip list
9. PyCharm、Jupyter Notebook 等
10. Tab
11. Alt + 3
12. Alt + 4
13. 编辑模式、命令模式
14. 可变类型、不可变类型
15. 字符串、列表、元组
16. 整数、浮点数、复数
17. True、False
18. 字典
19. 二进制、八进制、十进制、十六进制
20. 0b 或 0B、0o 或 0O、0x 或 0X
21. 22
22. '0b11'或'0B11'
23. '0o13'或'0O13'
24. '0x14'或'0X14'
25. 带小数点的一般形式、科学记数法

- 26. 2.0
- 27. 2
- 28. 2
- 29. 3.142
- 30. 5
- 31. j、J
- 32. x.real、x.imag
- 33. $4 + 3j$ 或 $4 + 3J$
- 34. 5.0
- 35. 前后顺序、相同的
- 36. 单行字符串、多行字符串
- 37. 单引号、双引号
- 38. 三个单引号、三个双引号
- 39. 0、-1
- 40. 1
- 41. 'n'
- 42. 'th'
- 43. 'Pto'
- 44. "
- 45. "
- 46. 'yh'
- 47. 'Python'
- 48. 'dog'
- 49. 字符串、元组
- 50. type()
- 51. '5'
- 52. 单行注释、多行注释
- 53. 5.0
- 54. 5.0
- 55. 抛出异常
- 56. 1
- 57. 5
- 58. 抛出异常
- 59. '(4+3j)'
- 60. '1.5'
- 61. 转义字符或反斜杠或“\”
- 62. 转义字符

- 63. 转义字符或反斜杠或“\”
- 64. \n、\r、\t、\v、\f 等
- 65. UTF-8、UTF-16、UTF-32
- 66. '0o101'
- 67. 元素或项
- 68. numbers.append(5)
- 69. [1, 2, 1, 2]
- 70. [1, 2, 3]
- 71. 2
- 72. 1
- 73. [3, 2, 1]
- 74. [1, 3]
- 75. [2, 3]
- 76. 0
- 77. 3
- 78. ['a', 'b']
- 79. t.extend([3, 1])
- 80. t.append([3, 1])
- 81. t.insert(1, 5)
- 82. t.reverse()
- 83. [1, 2, 5]
- 84. [5, 2, 1]
- 85. [1, 5, 2]
- 86. 不可变的
- 87. (5,)
- 88. (2, 1, 2, 1)
- 89. ()
- 90. (1, 5)
- 91. ('a', 'b')
- 92. 元组
- 93. a, b = b, a
- 94. sort()
- 95. False
- 96. False
- 97. False
- 98. True
- 99. False

- 100. 'aaa'
- 101. 成员资格测试、消除重复元素
- 102. {1}
- 103. {2, 3}
- 104. st.add(2)
- 105. {1, 2}
- 106. True
- 107. {}
- 108. dt["a"] = 3
- 109. ['b', 'a']或['a','b']
- 110. ['b', 'a']或['a','b']
- 111. [3, 1]或[1,3]
- 112. [('b', 3), ('a', 1)]或[('a', 1), ('b', 3)]
- 113. 'not exists'
- 114. 1
- 115. dt 不变,即 dt = {'b': 3, 'a': 1}
- 116. {'b': 3, 'a': 1, 'c': 2}
- 117. 2
- 118. {'a': 1}
- 119. 1=2=3 #
- 120. 3
- 121. '5'
- 122. input()、print()
- 123. 2.0
- 124. 1
- 125. 3
- 126. -3
- 127. -3
- 128. 2
- 129. 2.5
- 130. 8
- 131. 0.25
- 132. 1
- 133. False
- 134. and、or、not、not
- 135. False
- 136. True

- 137. False
- 138. False
- 139. 5
- 140. hello
- 141. False
- 142. 2
- 143. 8
- 144. 6
- 145. 14
- 146. -13
- 147. 3
- 148. 11110011
- 149. False
- 150. False
- 151. True
- 152. 选择、循环
- 153. while、for
- 154. 多分支结构
- 155. 循环次数已知的情况
- 156. [1, 3]
- 157. (0, 1, 2, 3, 4)
- 158. {1, 2, 3, 4}
- 159. else
- 160. [0, 1, 4]
- 161. def
- 162. 单星操作符 *、双星操作符 **
- 163. 默认值
- 164. 必选、可选
- 165. 按位置赋值、按关键字赋值
- 166. 终止条件、递归条件
- 167. lambda
- 168. 作用域
- 169. 局部变量、全局变量
- 170. global
- 171. class
- 172. 抽象或概括
- 173. __init__()

- 174. 实例、类
- 175. 实例
- 176. `type()`
- 177. 是否属于类型 `T`
- 178. 多重继承
- 179. 多态
- 180. 封装、继承、多态
- 181. 公共属性
- 182. `@classmethod`
- 183. `@staticmethod`
- 184. `super()`
- 185. `__add__()`
- 186. `import`
- 187. 一个扩展名为 `py` 的文件
- 188. 一个文件夹、`__init__.py`
- 189. `as`
- 190. `__init__.py`
- 191. `random`
- 192. `random.seed(42)`
- 193. `choice()`
- 194. `[0, 1)`
- 195. `0, 1`
- 196. 随机选择并返回 `x` 中的 `k` 个元素
- 197. `time`
- 198. `sleep()`
- 199. `time()`
- 200. 四位数的年份
- 201. `0`
- 202. `'god'`
- 203. `'ONE'`
- 204. `False`
- 205. `True`
- 206. `'one'`
- 207. `'ab'`
- 208. `5`
- 209. `'a=b'`
- 210. `32`

- 211. re
- 212. 普通字符
- 213. 元字符
- 214. \w
- 215. \d
- 216. 空白
- 217. t、d
- 218. ^
- 219. \b
- 220. +
- 221. {m,n}
- 222. {m}
- 223. group()
- 224. re.I
- 225. re.M
- 226. re.S
- 227. ['a', 'b', 'c']
- 228. ['a', 'b', 'c']
- 229. 异常
- 230. 抛出异常
- 231. except
- 232. finally
- 233. raise
- 234. 集成
- 235. I/O 操作错误
- 236. unittest
- 237. 可见
- 238. open()
- 239. rt
- 240. with
- 241. seek()
- 242. tell()
- 243. wb+ 或 rb+
- 244. 美国标准信息交换码或 American Standard Code for Information Interchange
- 245. 1~4
- 246. closed 或 mode 或 name
- 247. 文件末尾或 End of File

- 248. 最多读取并返回 n 个字符。如果 n 为负数或空,则读取文件的所有内容
- 249. read()、readline()、readlines()
- 250. write()、writelines()
- 251. scandir()
- 252. 创建文件夹
- 253. 既可以创建单个文件夹,又可以创建目录树
- 254. 遍历目录
- 255. tempfile
- 256. os.remove()或 os.unlink()
- 257. os.rmdir()
- 258. os.getcwd()
- 259. os.chdir()
- 260. pickle、shelve、marshal
- 261. 大量数据的
- 262. sqlite3
- 263. 结构化查询语言或 Structured Query Language
- 264. SELECT
- 265. WHERE
- 266. INSERT INTO
- 267. UPDATE
- 268. DELETE
- 269. JOIN
- 270. GROUP BY
- 271. commit()
- 272. count()
- 273. 执行相关操作
- 274. SQLite
- 275. connect()、cursor()、execute()
- 276. 父容器
- 277. Button、Canvas、Checkbutton
- 278. font
- 279. c、i、m、p
- 280. 内边距、外边距
- 281. anchor
- 282. cursor
- 283. pack、grid、place
- 284. 按下任意键

- 285. 文本框或 Entry
- 286. 标签或 Label
- 287. 延迟 5ms 后,调用 callback 函数
- 288. Text
- 289. [True, '0', 'a']
- 290. [1, 4, 7]
- 291. 9
- 292. [(1, 'b')]
- 293. [(0, 'a')]
- 294. '中国梦'
- 295. '0.62'
- 296. 2
- 297. [2, 1]
- 298. 生成器表达式、生成器函数
- 299. 可迭代对象
- 300. arange()、linspace()、logspace()
- 301. 5
- 302. 10、15、20、25
- 303. 12
- 304. 0 或 1
- 305. (2, 2)
- 306. resize()或 reshape()或 ravel()
- 307. 将数组 a 和 b 水平堆叠
- 308. hsplit()
- 309. array_split()
- 310. Series、DataFrame
- 311. 2、2
- 312. 异构
- 313. hist()
- 314. plot()
- 315. show()
- 316. 样式字符
- 317. text()
- 318. ONE
- 319. lie
- 320. s[int(len(s) / 2)]或 s[len(s) // 2]
- 321. 'Not Exists'

- 322. -1
- 323. IDLE、pip
- 324. seek()、tell()
- 325. F5 键
- 326. Package Installer for Python
- 327. 命令提示符 cmd、Scripts
- 328. sys.executable
- 329. 基本型、组合型
- 330. 5
- 331. 17
- 332. 42
- 333. 60
- 334. 0.5
- 335. '0b11'
- 336. '0o12'
- 337. '0xc'
- 338. round()
- 339. 列表、元组、集合、字典
- 340. 1.23
- 341. a.real、a.image
- 342. 5.0
- 343. 1.5
- 344. 'e'
- 345. 'dog'
- 346. str()
- 347. 5
- 348. '11'
- 349. 8
- 350. 'haha'
- 351. 'haha'
- 352. 'goodidea'
- 353. 3
- 354. [1, 2, 3, 1]
- 355. (1, 2, 3, 1)
- 356. ['b', 'd', 'a']
- 357. ['a', 'd', 'b']
- 358. ['d', 'a']

- 359. ['d']
- 360. ['b', 'e']
- 361. ['g', 'o', 'd']
- 362. ['amazing', 'China']
- 363. 'just=for=fun'
- 364. 'C-C-T-V'
- 365. 'a'
- 366. ('o', 'n', 'e')
- 367. 2
- 368. 0
- 369. 'qq.com'
- 370. {'n', 'a', 'b'}, 元素的顺序无关紧要
- 371. hash()
- 372. 字符串
- 373. 4.6
- 374. 5
- 375. Just-for-Fun
- 376. 1
- 377. 8
- 378. 12
- 379. -13
- 380. 72
- 381. 15
- 382. g-o-o-d-
- 383. [0, 1, 2, 3, 4]
- 384. object 类
- 385. "
- 386. False
- 387. '26'
- 388. False
- 389. 'c'
- 390. 'g-o-o-d'
- 391. 'a blue blue rose'
- 392. 'fun'
- 393. ['one', 'two', 'three']
- 394. ['1', '3']
- 395. WHERE 子句

- 396. background 或 bg、foreground 或 fg
- 397. [6, 7, 8, 9]
- 398. 6
- 399. [2, 4]
- 400. [(0, 'a'), (1, 'b')]
- 401. 3
- 402. 5
- 403. in
- 404. 低耦合、高内聚
- 405. 数据约束
- 406. 物理模型

3.2 单选题参考答案

- 1-5 D A D B B
- 6-10 C B A A C
- 11-15 D C C D D
- 16-20 D D D D B
- 21-25 D D D C D
- 26-30 B A B B A
- 31-35 B D C B D
- 36-40 B A A C C
- 41-45 D C D D A
- 46-50 C B D A A
- 51-55 D D C C C
- 56-60 D B D D D

3.3 简答题参考答案

- 1. 对浮点数 x 进行四舍五入,保留 d 位小数。
- 2. 安装 module 模块: pip install module
查看本机已安装的所有模块: pip list
升级 module 模块: pip install --upgrade module
- 3. Python Package Index。PyPI 是 Python 语言的软件库。
- 4. pyinstaller -F demo.py
- 5. 5、6
- 6. 两种,分别是单行字符串和多行字符串。

单行字符串用一对单引号(')或一对双引号(")作为边界;

多行字符串用一对三单引号('')或一对三双引号('"')作为边界。

7. 两种,分别是单行注释和多行注释。

8. 以一个反斜杠“\”开头的字符序列。

退格符\b,换行符\n,回车符\r,水平制表符\t。

9. 反斜杠(\)也是续行符,也就是将一行代码写成两行或两行以上,一般用于代码较长的行,以增加可读性。

10. UTF-8、UTF-16 和 UTF-32。

11. 只能使用 52 个大小写英文字母、0 ~ 9 十个阿拉伯数字和一个下画线“_”;

变量名不能以数字开头;

不能用 Python 解释器使用的关键字(Keyword)。

12. 两种,分别是 list()和[]。

13. 中括号运算符[]。

14. 3 种,分别是 pop()、del 和 remove()。

15. 'b'、['a', 'c']。

16. 元组是不可改变的(Immutable)。

17. 两种,分别是 tuple()和()。

18. 一种,set()。

19. 两种,分别是 dict()和{ }。

20. {'g', 'f'}或{'f', 'g'}。

21. ['b', 'a', 'c']。

22. input()函数。

23. $3 / 2 = 1.5$

24. (1) 3.5 (6) 1

(2) 3 (7) 5.0

(3) -4 (8) True

(4) (3, 1) (9) False

(5) 9 (10) False

25. 10001101、11110010、11110011

26. 两个,分别是 is 和 is not。

27. 两种,分别是选择结构和循环结构。

28. 两种,分别是 for 循环和 while 循环。

29. for 和 while 循环分别适用于循环次数已知和循环次数未知的情况。

30. 1 2 3 4。

31. 4

6

7

else 分支

- 32. 两种,分别是局部变量和全局变量。
- 33. 两种,分别由单星 * 操作符和双星**操作符定义。

34. class

- 35. 两种,分别是类方法和实例方法。
- 36. 两种,分别是类属性和实例属性。

37. 多重继承(Multiple Inheritance)

38. 3种,分别是私有成员、保护成员和公共成员。

公共成员能在类的外部访问;保护成员可以在该成员所在类及其子类中访问;私有成员只能在成员所在类中使用。

39. 如果一个父类及其子类使用了完全相同的方法名,但有不同的实现方式,这种现象叫作多态(Polymorphism)。

40. __new__()构造方法,创建实例时自动调用;

__init__()实例的初始化方法;

__del__()析构方法,释放实例时自动调用。

41. 这两种方法几乎是等价的。唯一的区别:如果找不到子串,index()方法抛出异常,而 find()方法返回-1。

42. 字符串 s 的值保持不变,即 s = 'fun\t\n'。

43. 点“.”、匹配行首“^”、匹配行尾“\$”。

44. [' ', 'a', 'b', 'c']。

注意: 结果列表的第一个元素为空。

45. try-except 语句和 raise 语句。

46. finally 子句和 else 子句。

47. BaseException 类。

48. 两种,分别是单元测试(Unit Test)和集成测试(Integration Test)。

49. 无论是否发生异常情况,都需要执行一些清理操作的地方,比如关闭文件。

50. open('test.dat', 'wb+')

51. 空字符串"。注意不是空格',也不是空行\n。

52. close()、read()、write()、tell()。

53. open()是 Python 语言的一个内置函数,用于打开文件。

54. r 只读模式(默认)、w 只写模式、a 追加模式。

55. 使用 try...finally 异常处理结构或者使用 with 语句。

56. pickle 模块和 shelve 模块。

57. tempfile 模块。

58. os 模块、shutil 模块和 pathlib 模块。

59.

import os

```
print(os.getcwd())
```

60. SQL 是 Structured Query Language 的缩写,它是一种结构化查询语言。使用 SQL 可以实现与数据库交互。

61. 一般流程包括 5 个步骤:

- ① 创建数据库连接 connection;
- ② 获取游标 cursor;
- ③ 执行相关操作;
- ④ 关闭游标 cursor;
- ⑤ 关闭数据库连接 connection。

62.

```
import sqlite3
from sqlite3 import Error
try:
    connection = sqlite3.connect('test.sqlite')
except Error as e:
    print(e)
finally:
    connection.close()
```

63.

```
create_users_table = """CREATE TABLE IF NOT EXISTS users(
ID INTEGER PRIMARY KEY AUTOINCREMENT,
NAME TEXT NOT NULL,
GENDER TEXT NOT NULL,
AGE INTEGER NOT NULL
);"""
```

64. cursor()用于创建并返回一个游标对象、commit()用于提交当前事务、close()用于关闭连接。

65. execute()用于执行一条 SQL 语句、fetchone()用于获取查询结果集的下一行、fetchall()用于获取查询结果集的所有行、close()用于关闭游标对象。

66. PRIMARY KEY 主键;FOREIGN KEY 外键。

67. DB Browser (SQLite)。

68. 连接操作。

69. Update 和 Delete。

70. SELECT * FROM users

71. 4 个步骤:

- ① 加载 tkinter 模块;
- ② 创建主窗口;

③ 在应用程序中添加一个或多个组件,比如按钮 Button;

④ 进入主事件循环。

72. import tkinter 或 from tkinter import *

73. Button 按钮组件、Canvas 画布组件、Checkbutton 复选按钮、Label 标签组件、Text 文本组件。

74. borderwidth 或 bd、height、width。

75. 两种,分别为

(1) 可以使用字符串,指定十六进制数字中 RGB 的比例。比如,"#fff"表示白色、"#000000"表示黑色、"#000fff000"表示纯绿色。

(2) 可以使用任何本地定义的标准颜色名称,比如白色"white"。

76. 两种,分别为

(1) 作为字体属性使用。

首先生成一个字体实例:

```
import tkinter.font as tkFont
ft = tkFont.Font(family="隶书", size=14, weight=tkFont.BOLD, slant=tkFont.ITALIC)
```

然后将其作为字体属性使用:

```
label = Label(root, text="字体设置", font=ft)
```

在上述代码中,字体实例 ft 被作为标签组件的字体属性 font 使用。

(2) 以字体三元组的方式使用。

其实第一种使用方式,完全可以用一行代码代替:

```
label = Label(root, text="字体设置", font=("隶书", 14, "bold italic"))
```

上述代码中的 font=("隶书", 14, "bold italic"),就是一个字体三元组。此处 font 是一个元组,其由 3 个元素构成。

77. anchor;用于定义文字放置位置的参照点。

78. relief 属性。常用的边框样式有 FLAT、SUNKEN、RAISED。

79. 3 种,分别是 pack 类、grid 类和 place 类。

80. 3 个属性,分别是 side 属性、expand 属性和 fill 属性。

81. sum = 3。

82. 2

83.

```
one:two:three
one-two-three
```

sep 是可选参数/关键字参数。

84.

4

24

36

85.

10

5

6

86. 2

87. 3

88. else 分支

89. `__name__` 变量是 Python 的内置变量。每个 Python 模块都有一个 `__name__` 变量。如果程序被作为模块加载,则其 `__name__` 变量的值被自动设置为模块名;如果程序是独立运行的,则其 `__name__` 属性值被自动设置为 `__main__`。利用 `__name__` 变量可控制 Python 程序的运行方式。

90. 两种功能,分别是位运算和集合的交集运算。

```
>>> 0b101 & 0b011
1
>>> 0b101 & 0b110
4
>>> st1 = {1, 5, 3}
>>> st2 = {3, 2, 4}
>>> st1 & st2
{3}
```

91. (1) 二叉树如图 3-1 所示。

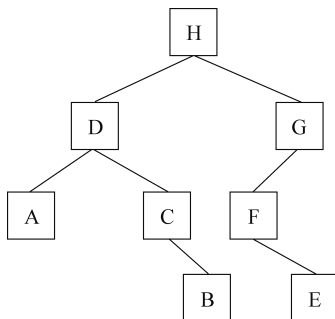


图 3-1 二叉树

(2) 后序序列: ABCDEFGH。

92. 迭代器是一次性的,而可迭代对象能执行无数次迭代循环。

93.

```
>>> class MyIterable:
    def __iter__(self):
        pass
>>> it = MyIterable()
>>> isinstance(it, Iterable)
True
>>> isinstance(it, Iterator)
False
```

94. 两种,分别是生成器表达式和生成器函数。

95.

```
>>> import sys
>>> gen = (i * 2 for i in range(1000))
>>> sys.getsizeof(gen)
120
```

96. iter(obj)

97.

```
a + b = array([[5, 4],
               [4, 3]])
```

98. array()、arange()、linspace()和 logspace()

99. Series 和 DataFrame

100. 一个函数在其函数体内部调用它自身,这种函数叫作递归函数。递归函数由终止条件和递归条件两部分构成。

3.4 编程题参考答案

1. a, b = b, a

2.

```
user_name = input("输入您的姓名:")
print("Hello, ", user_name)
```

3.

```
num1 = int(input("输入第一个数:"))
num2 = int(input("输入第二个数:"))
print(num1 + num2)
```

4.

```
num1 = int(input("输入第一个数:"))
```

```
num2 = int(input("输入第二个数:"))
if num1 > num2:
    print("第一个数大:", num1)
else:
    print("第二个数大:", num2)
```

5.

```
n1 = int(input("输入第一个数:"))
n2 = int(input("输入第二个数:"))
n3 = int(input("输入第三个数:"))
if n1 >= n2 and n1 >= n3:
    print("第一个数最大:", n1)
elif n2 >= n1 and n2 >= n3:
    print("第二个数最大:", n2)
else:
    print("第三个数最大:", n3)
```

或者

```
n1 = int(input("输入第一个数:"))
n2 = int(input("输入第二个数:"))
n3 = int(input("输入第三个数:"))
print(max(n1, n2, n3))
```

6.

```
for num in range(2, 100, 2):
    print(num)
```

7.

```
for num in range(50, 0, -1):
    if num % 2 == 1:
        print(num)
```

8.

```
import math
def is_prime(num):
    divisor = int(math.sqrt(num)) + 1

    for i in range(2, divisor):
        if num % i == 0:
            return False
    return True

for num in range(2, 100):
```

```
if is_prime(num):  
    print(num)
```

9.

```
def mul_or_sum(num1, num2):          #mul 代表 multiplication  
    product = num1 * num2  
    if product <= 1000:  
        return product  
    else:  
        return num1 + num2  
num1 = 30  
num2 = 20  
result = mul_or_sum(num1, num2)  
print("The result is:", result)
```

一个更简洁的答案如下:

```
a, b = int(input("输入第一个整数:")), int(input("输入第二个整数:"))  
prod = a * b  
print(prod if prod <= 1000 else a + b)
```

10.

```
user_str = 'hello'  
for i in range(0, len(user_str), 2):  
    print(user_str[i])
```

一个更简洁的答案如下:

```
user_str = input("输入一个字符串:")  
print(user_str[0::2])
```

11.

```
def remove_chars(str, n):  
    return str[n+1:]  
print(remove_chars('hello', 2))
```

12.

```
def first_last_is_same(number_list):  
    first_element = number_list[0]  
    last_element = number_list[-1]  
    if first_element == last_element:  
        return True  
    else:  
        return False  
result = first_last_is_same([10, 20, 30, 40, 10])
```