

第 1 章 Python 概述

一、单选题

1	2	3	4	5	6	7
C	B	A	D	C	B	A

二、填空题

- 对象
- 可移植性
- pip、setuptools
- quit()、Ctrl+Z
- F5
- #
- sys.argv、argv[0]、argv[1]
- help()或者 help; quit

第 2 章 Python 语言基础

一、单选题

1	2	3	4	5	6	7	8	9	10	11
A	A	B	D	B	C	C	A	B	A	C

二、填空题

- 简单语句
- 缩进对齐
- \
- ;
- #
- pass
- 2**32-1
- 2.25、2.0、0.5
- 1.0
- 整数类型 int、字符串 str、complex、元组 tuple、字节序列 bytes。列表 list、字典 dict、集合 set、字节数组 bytearray
- is 和 is not、type()、==
- 4 3

三、思考题

- 20、8、50、2.0、2、1
- 9 8 7 6
- <class 'NoneType'>
- True True False True True。说明：

```
x = y = [1, 2] #变量 x 和 y 指向 list 对象[1, 2]
x.append(3) #变量 x 指向的 list 对象[1, 2]附加一个元素
x is y #输出: True。表示变量 x 和 y 指向同一个 list 对象[1, 2, 3]
x == y #输出: True。表示变量 x 和 y 指向的 list 对象值相等
z = [1, 2, 3] #变量 z 指向的 list 对象[1, 2, 3]
x is z #输出: False。表示变量 x 和 z 指向不同的 list 对象[1, 2, 3]
x == z #输出: True。表示变量 x 和 z 指向的 list 对象值相等
y == z #输出: True。表示变量 x 和 z 指向的 list 对象值相等
```

第 3 章 程序流程控制

一、单选题

1	2	3	4	5	6	7	8	9
A	A	B	A	B	D	C	B	A

二、填空题

- __iter__()、__next__()
- break
- 1 6 11 16
- 10 8 6 4 2
- 6
- 25 或者 26
- 1、1

三、思考题

- (1) if(i > 0):
 if(j > 0): n = 1
 else: n = 2

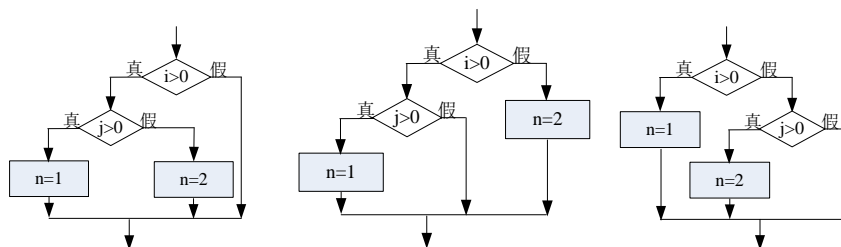
相当于： $\begin{cases} i > 0, j > 0 & n = 1 \\ i > 0, j \leq 0 & n = 2 \end{cases}$ ，流程图参见图 3-1(a)所示。

(2) if (i > 0):
 if (j > 0): n = 1
 else: n = 2

相当于： $\begin{cases} i > 0, j > 0 & n = 1 \\ i \leq 0 & n = 2 \end{cases}$ ，流程图参见图 3-1(b)所示。

(3) if (i > 0): n = 1
 else:
 if (j > 0): n = 2

相当于： $\begin{cases} i > 0 & n = 1 \\ i \leq 0, j > 0 & n = 2 \end{cases}$ ，流程图参见图 3-1(c)所示。



(a)思考题 1 (1) (b)思考题 1 (2) (c)思考题 1 (3)

图 3-1 思考题 1 流程图

2. 0 1 2 2 3 4

3. 打印 100~200 间的全部素数，每行输出 10 个。

4. 利用循环语句显示有规律的图形（用*构成的上三角），要求输入显示的行数。运行效果如图 3-2 所示。

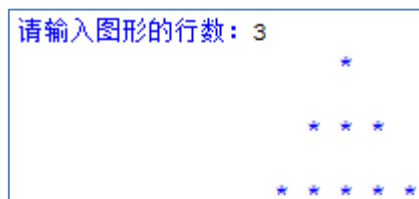


图 3-2 用*构成的上三角运行效果

5. 本题显示三位数中的所有水仙花数，运行效果如图 3-3 所示。所谓“水仙花数”是指一个三位数，其各位数字立方和等于该数字本身。例如，153 是水仙花数，因为 $153=1^3+5^3+3^3$ 。

三位数中的所有水仙花数为：
153 370 371 407

图 3-3 水仙花数运行效果

6. 本题找出 1000 以内的所有完数，运行效果如图 3-4 所示。一个数如果恰好等于它的因子之和，这个数就称为“完数”。例如，6 的因子为 1、2、3，而 $6=1+2+3$ ，因此 6 就是“完数”。

```
1~1000之间所有的完数有，其因子为：
6: [1, 2, 3]
28: [1, 2, 4, 7, 14]
496: [1, 2, 4, 8, 16, 31, 62, 124, 248]
```

图 3-4 1000 以内的所有完数运行效果

7. 求任意两个整数的最大公约数。

第 4 章 常用内置数据类型

一、单选题

1	2	3	4	5	6	7	8	9	10	11
A	A	C	A	B	D	A	D	C	D	B

二、填空题

- 整数类型 (int)、布尔类型 (bool)、浮点类型 (float)、复数类型 (complex)
- 元组 (tuple)、列表 (list)、字符串 (str) 和字节数据 (bytes 和 bytearray)
- 10 4. 6561 5. 1.8888888888888888 6. True 7. 9.0 18.7 20.0
- 124.0 100.0 15 9. 32 5 10. 0x10 0b1010 11. (5, 2)
- (3, 2) 13. 4 (2, 1) 14. (-5+12j) 15. 6 16
- 3.2 2.23606797749979 17. random
- $\text{math.sin}(15*\text{math.pi}/180)+(\text{math.pow}(\text{math.e},x)-5*x)/\text{math.sqrt}(x*x+1)-\text{math.log}(3*x)$
- $((c*d)/2/(c+d)-4*\text{math.pi}/(c-d))/(a+b)$ 20. Dict 21. True 22. False
- $x>0$ and $y>0$ or $x<0$ and $y>0$ 24. $i \% 3 == 0$ and $i \% 5 == 0$ 25. True
- True 27. True False 28. B 29. helloworld

三、思考题

- 1.78、1.8、1.7、1.7、-1.5、-1.5、1.7、1.6、-1.5、1.6、-1.4
- 随机产生一个 3 位正整数，然后逆序输出。
- 数量 100，单价 285.6、数量 100，单价 285.60、数量 100，单价 285.600
- ```

1
121
12321

```
- no            10. 2
- 1 2 True True 2、2 0 False 2 False。说明：(1)  $C = A \text{ or } B$ 。如果 A 不为 0 或不为空或为 True，则返回 A；否则返回 B。仅在必要时才计算第二个操作数，即如果 A 不为 0 或不为空或为 True，则不用计算 B。(2)  $C = A \text{ and } B$ 。如果 A 为 0 或为空或为 False，则返回 A；否则返回 B。仅在必要时才计算第二个操作数，即如果 A 为 0 或为空或为 False，则不用计算 B。
- T T F T F T F T F T。说明：如果 Python 表达式的结果为数值类型 (0)、空字符串 ("")、空元组 ()、空列表 []、空字典 {}，则其 bool 值为 False (假)；否则其 bool 值为 True (真)。例如：123、"abc"、(1,2)均为 True。

## 第 5 章 系列数据类型

### 一、单选题

|   |   |   |   |   |   |   |   |   |    |    |    |
|---|---|---|---|---|---|---|---|---|----|----|----|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | 9 | 10 | 11 | 12 |
| D | A | C | C | C | D | C | C | D | D  | A  | C  |

## 二、填空题

1. r                      2. 0                      3. False                      4. 45                      5. 1%1                      6. 5
7. 7                      8. (0, 1) [0, 1]                      9. (1, 2, 3) [1, 2, 3]                      10. [1, 3]、[0, 1, 4]
11. [1, 2, 3, 4]、[3, 4, 5]、[0, 1, 2]、90.0
12. 'c'、('c', 'd')、('a', 'b', 'c')、('d', 'e')、('b', 'd')、'd'、('e', 'd', 'c', 'b', 'a')、('d',)、('d', 'e')、()、('a', 'b')、('a', 'b', 'c', 'd', 'e')、('b', 'c', 'd')
13. 6、1、[4, 'x', 'y']                      14. [[1, 2], 7, 'a']

## 三、思考题

2. 利用循环语句显示有规律的图形，要求输入上(或下)三角的行数。运行效果如图图 5-1 所示。



图 5-1 沙漏型三角形运行效果

3. 利用循环语句显示有规律的图形，要求输入上(或下)三角的行数，运行效果如图图 5-2 所示。

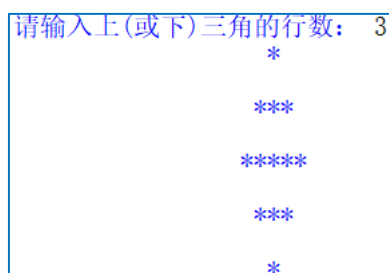


图 5-2 菱形三角形运行效果

4. 先输出星期,然后输出月份。即: DAYS: ['Monday', 'Tuesday', 'Wednesday', 'Thursday', 'Friday', 'Saturday', 'Sunday'], MONTHS ['Jan', 'Feb', 'Mar', 'Apr', 'May', 'Jun', 'Jul', 'Aug', 'Sep', 'Oct', 'Nov', 'Dec']
5. 输出结果是 4。说明: 语句 names2=names1 使得 names1 和 names2 指向(引用)相同的对象实例,而语句 names3=names1[:]表示 names3 是复制(创建)的新实例,其内容与 names1 (也即 names2) 的内容相同。

## 第 6 章 输入和输出

### 二、填空题

1. 1-2-3-4-5!                      2. 0 1 2 3 4 5 6 7 8 9                      3. sys.argv、argv[0]、argv[1]、argv[2]
4. argparse                      5. getpass                      6. with                      7. sys.stdin、sys.stdout 和 sys.stderr

## 第 7 章 错误和异常处理

### 一、单选题

|   |   |   |   |   |   |   |
|---|---|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 | 6 | 7 |
| B | D | B | D | B | C | C |

## 二、填空题

1. try、except、finally      2. raise      3. assert      4. -O、False      5. Exception

# 第 8 章 函数和函数式编程

## 一、单选题

|   |   |   |   |   |
|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 |
| D | D | D | C | C |

## 二、填空题

1. 5.5      2. global      3. 全局变量、局部变量和类型成员变量  
4. getrecursionlimit 和 setrecursionlimit      5. globals()和 locals()

## 三、思考题

4. 24      5. 1 4 9      6. simple function      7. 4 0  
8. 求两个数的最大公约数。输出结果为：3、None      9. quick  
10. <class 'tuple'>、(2, 3, 4, 5)。在声明函数时，通过带星的参数，如\*param2，允许向函数传递可变数量的实参。调用函数时，从那一点后所有的参数被收集为一个元组。  
11. <class 'dict'>、{'a': 2, 'b': 3, 'c': 4, 'd': 5}。在声明函数时，通过带双星的参数，如\*\*param2，允许向函数传递可变数量的实参。调用函数时，从那一点后所有的参数被收集为一个字典。带星或双星的参数必须位于形参列表的最后两个位置。

# 第 9 章 面向对象的程序设计

## 一、填空题

1. 封装、继承和多态      2. False      3. .      4. \_\_new\_\_、\_\_init\_\_、\_\_del\_\_      5. self

## 二、思考题

3. 100 100      4. 100      5. 400      6. 12      7. 7  
8. 7      9. 30      10. 16  
11. 21。“object.\_\_dict\_\_”返回对象的属性字典，本题为{'id': 123, 'age': 18, 'gender': 'female'}。

# 第 10 章 模块和客户端

## 一、填空题

1. import      2. from m import \*      3. \_\_import\_\_()      4. sys.path  
5. \_\_name\_\_、\_\_main\_\_      6. dir()、help()

# 第 11 章 算法与数据结构基础

## 一、单选题

|   |   |   |   |   |
|---|---|---|---|---|
| 1 | 2 | 3 | 4 | 5 |
| B | D | A | C | C |

## 二、填空题

1. 0      2. 2      3. {1, 2, 3}      4. 15      5. 食品  
6. g      7. {1, 2, 3, 5}、{2, 3}、{1}、{1, 5}、{1, 2, 3, 'x'}、{2, 3}、{1, 2}、set()  
8.  $n^2-1$ 、 $O(n^2)$       9. sys.getsizeof(x)      10. 逻辑、存储、运算

## 三、思考题

1. 6      2. {'x': 3, 'c': }      3. {'Pear': 1, 'Apple': 2, 'kiwi': 1, 'apple': 2}  
4. 输出结果是：3      6      {(1, 2): 3, (2, 1): 2, (1, 2, 3): 1}

5. 输出结果是：12。语句 `d2= d1` 使得 `d1` 和 `d2` 指向（引用）相同的对象实例。
6. 输出结果是：7。本题中的 `d2= dict(d1)` 相当于 `d2= d1.copy()`。
7. `[{1: 'a', 2: 'b'}, {2: 'a', 3: 'x', 4: 'y'}]` `ChainMap({2: 'a', 3: 'x', 4: 'y'})` `ChainMap({}, {1: 'a', 2: 'b'}, {2: 'a', 3: 'x', 4: 'y'})`、`a x`、`ChainMap({1: 'A', 2: 'b', 3: 'X'}, {2: 'a', 3: 'x', 4: 'y'})`
8. `Counter()`、`Counter({'a': 3, 'n': 2, 'b': 1})`、`Counter({'R': 4, 'B': 2})`、`Counter({'dogs': 8, 'cats': 4, 'birds': 2})`、0 4、`['R', 'R', 'R', 'R', 'B', 'B']`、`[('dogs', 8), ('cats', 4)]`、`Counter({'R': 3, 'B': 1, 'G': -1})`
9. `a 2 c c 2 a`                      10. `a 2 c`                      11. `1 0`、`[('r', [3, 5]), ('b', [1, 4]), ('g', [2])]`
12. `dict_items([('red', 3), ('green', 4), ('blue', 1)])` `[('blue', 1), ('green', 4), ('red', 3)]`、`('red', 3)` `('blue', 1)`
13. `('x', 'y')` 1 2、`OrderedDict([('x', 10), ('y', 20)])`、`Point(x=100, y=20)` 1 2
14. `array('i', [1, 22, 3, 4, 5])` `array('i', [3, 4, 5])` `<class 'int'>`、`array('i', [1, 22])` `i 4`
15. `array('b', [3, 2, 3, 3, 5])` 3、`array('b', [3, 2, 3, 3, 5, 65, 49, 8, 9])` 0、`[8, 49, 65, 5, 3, 3, 3, 1]`

## 第 12 章 图形用户界面

### 一、填空题

1. `_tkinter`、`tkinter` 和 `tkinter.constants`                      2. 根窗口或主窗口    3. `pack`、`grid` 和 `place`
4. `width` 和 `height`    5. `font`    6. `anchor`                      7. `cursor`
8. `text`、`wrlength`、`justify`                      9. `bitmap`、`.xbm`                      10. `image`    11. `compound`
12. `relief`、`overrelief`    13. `borderwidth` 或 `bd`                      14. `padx` 和 `pady`    15. `state`
16. `underline`                      17. `textvariable`                      18. `Radiobutton`（单选按钮）
19. `Checkbutton`（复选框）    20. `Listbox`（列表框）                      21. `OptionMenu`（选择项）
22. `Scale`（移动滑块）                      23. `messagebox`、`filedialog`、`colorchooser` 和 `simplifiedialog`
24. `messagebox`    25. `filedialog`    26. `colorchooser`                      27. `simplifiedialog`

## 第 13 章 图形绘制

1. `tkinter`、`turtle`    2. `Canvas`（画布）                      3. 左上角、右下角                      4. `Pyplot`

## 第 14 章 数值日期和时间处理

### 一、填空题

1. `datetime`、`calendar`、`time`                      2. `date`、`time`、`datetime`、`timedelta`、`tzinfo`、`timezone`
3. `gettime`                      4. `daylight`                      5. `strptime()`、`strftime()`
6. `datetime.MINYEAR` 和 `datetime.MAXYEAR`、1 和 9999
7. `strftime()`、`strptime()`                      8. `td.days`、`td.seconds`、`td.microseconds`                      9. `True`

### 二、思考题

1. `0001-01-01` `9999-12-31` `0001-02-01`、`2019 10 1`、`737333 1 Tue Oct 1 00:00:00 2019`  
`2019/10/01(Tue)`                      2. `00:00:00 23:59:59.999999`、`19 30 45 196`、`19 时 30 分 45 秒`
3. `0 0001-01-01 00:00:00 9999-12-31 23:59:59.999999`、`2019 5 1 9 35 46`、`2019-05-01 09:35:46`  
`2019/05/01(Wednesday)`、`09 时 35 分 46 秒`
4. `0:25:00 300 1:40:00 True`                      5. `31 2019-06-11 2019-05-22 True`

## 第 15 章 字符串和文本处理

### 一、填空题

1. `RED HAT`、`'RED HAT'`、`'Red Hat'`、`'red cat'`
2. `'0000abc'`、`' abc '`、`'abc '`、`'0000abc'`

3. ['a', 'b', 'c']、['a,b', 'c']、('a', ',', 'b,c')、('a,b', ',', 'c')、'a:b:c'、'x:y:z'  
4. 空      5. ['to', 'to']      6. ['boy', 'box']      7. \d{6}      8. /i、/m  
9. 'Python is easy to learn.'      10. ['go', 'went', 'gone']      11. ['a', 'b', 'c', '']

## 二、思考题

5. 分行输出 fruits 列表中各元素的值，等价于输出“pear\napple\nkiwi\navocado\norange\n”。  
6. birth、happy Birthday  
7. 8. 注意，match 函数从字符串头部开始匹配，而 search 函数在字符串任何位置匹配。  
8. None、<re.Match object; span=(10, 12), match='to'>、['to', 'to']  
9. to (10, 12)  
10. c.isalpha() 或者：c.lower() <= 'z' and c.lower() >= 'a' 或者：  
c.upper() <= 'Z' and c.upper() >= 'A' 或者：c <= 'Z' and c >= 'A' or c <= 'z' and c >= 'a'  
11. c.isdigit() 或者：c <= '9' and c >= '0'  
12. c.isupper() 或者：c <= 'Z' and c >= 'A'  
13. c.islower() 或者：c <= 'z' and c >= 'a'

## 第 16 章 文件和数据交换

### 一、填空题

1. open      2. close、with      3. seek      4. fileinput  
5. pickle/cPickle/marshal

## 第 18 章 网络编程和通信

### 一、填空题

1. 网络接口层、Internet 层、传输层和应用层      2. IP 地址  
3. 域名系统（Domain Name System, DNS）  
4. 统一资源定位器（Uniform Resource Locator, URL）  
5. TCP、UDP      6. listen 和 accept      7. connect

## 第 19 章 并行计算：线程、进程和协程

1. start\_new\_thread、\_thread.exit()      2. run、start      3. 用户线程和 daemon 线程  
4. daemon 线程      5. locked 和 unlocked（初始状态）

## 第 20 章 系统管理

### 一、填空题

1. mkdir      2. tempfile      3. chdir      4. listdir      5. glob      6. walk  
7. join      8. exists      9. isfile      10. isdir      11. isabs      12. islink  
13. getatime      14. getmtime      15. getctime      16. getsize      17. remove      18. rmdir  
19. rmtree      20. copytree      21. move      22. copy      23. copy2  
24. disk\_usage      25. system      26. popen      27. get\_terminal\_size  
28. make\_archive 和 unpack\_archive      29. get\_archive\_formats 和 get\_unpack\_formats  
30. .ini 或 cfg      31. 节（Section）、键（Option）和值（Value）、=或:、#号或;号  
32. ConfigParser