

高级程序设计语言 同步测试

第一章 Python语言基础

判断题

1. Python是一种跨平台、开源、免费的高级动态编程语言。（对）
2. 不可以在同一台计算机上安装多个Python版本。（错）
在同一台计算机上可以安装多个Python版本，并且可以通过不同的方式进行管理，如anaconda。
3. 已知 `x = 3`，那么赋值语句 `x = 'abcdefg'` 是无法正常执行的。（错）
在Python中，变量是动态类型的，这意味着同一个变量可以在不同的时间存储不同类型的数据，下题同理。
4. Python变量使用前必须先声明，并且一旦声明就不能再当前作用域内改变其类型。（错）
5. Python不允许使用关键字作为变量名，允许使用内置函数名作为变量名，但这会改变函数名的含义。（对）
6. 只有Python扩展库才需要导入以后才能使用其中的对象，Python标准库不需要导入即可使用其中的所有对象和方法。（错）
Python标准库中的大部分模块和函数也是需要通过导入才能使用的。例如使用 `math` 模块中的 `sqrt` 函数，必须先导入 `math` 模块。
7. Python使用缩进来体现代码之间的逻辑关系。（对）

选择题

1. Python安装扩展库最常用的工具是（B）
A、Anaconda；B、pip；C、Pycharm；D、IDEL
A、Anaconda是一个Python的发行版，包含了Python及其常用的科学计算库、工具和包。它提供了一个便捷的环境管理工具conda，可以用来安装和管理Python包；B、pip是Python官方推荐的安装和管理第三方库的工具；C、PyCharm是一个集成开发环境；D、IDLE是Python自带的简单集成开发环境。
2. 下面用于导入Python标准库对象错误的语句是（C）
A、`import math as m`；B、`from math import sin`；C、`import math.*`；D、`from math import *`
A、它将 `math` 模块导入并且为它起一个别名 `m`；B、它将 `math` 模块中的 `sin` 函数导入到当前命名空间；C、Python不允许使用通配符 `*` 来导入模块的所有内容，通配符的使用方法是D选项。
3. 以下选项中，不是Python语音特点的是（B）
A、强制可读：Python语音通过强制缩进来体现语句间的逻辑关系；B、变量声明：Python语音具有使用变量需要先定义后使用的特点；C、平台无关：Python程序可以在任何安装了Python解释器的操作系统环境中执行；D、黏性扩展：Python语言能够集成C、C++等语言编写的代码
A、Python通过使用强制缩进来表示代码的逻辑结构，从而提高代码的可读性；B、Python变量是动态类型的，使用前不需要显式声明其类型，可以直接赋值和使用；C、Python程序是跨平台的，只要在目标操作系统上安装了Python解释器，程序就可以直接运行，同一段代码可以在Windows、Linux和macOS上运行；D、Python能够与其他语言编写的代码进行集成，如NumPy、Pandas等库底层部分使用C语言编写。
4. 下列关于Python的说法不正确的选项是（D）
A、正确的缩进对Python程序是非常重要的；B、在表达式中圆括号可以改变运算顺序；C、Python程序可以在任何安装了Python解释器的操作系统环境中执行；D、Python程序中的空格和空行可有可无，但适当地增加空格和空行可以提高代码的可读性
A、Python使用缩进来体现代码之间的逻辑关系；B、如 `result = (2 + 3) * 4` 就改变了运算顺序；C、Python是一种跨平台的语言，只要目标计算机上安装了相应版本的Python解释器，就可以运行Python程序；D、P不当的空格和空行可能导致语法错误或影响代码的执行。
5. 在Python3.x中，`input()`函数的返回值类型为（B）
A、list；B、str；C、int；D、object
返回字符串，但可以使用如 `int(input())` 转换成整型。
6. 下列关于数据分析的描述，说法错误的是（C）
A、模型优化步骤可以与分析和建模步骤同步进行；B、数据分析过程中最核心的步骤是分析与建模；C、数据分析时只能够使用数值型数据；D、广义的数据分析包括狭义分析和数据挖掘
C、文本数据可以使用自然语言处理技术进行分析，数据分析不局限于数值型数据。
7. 下列不属于数据分析的应用场景的是（D）
A、一周天气预测；B、合理预测航班座位需求数量；C、为用户提供个性化服务；D、某人一生的命运预测
显而易见。
8. 下列不属于Python优势的是（C）
A、语法简洁，程序开发速度快；B、入门简单，功能强大；C、程序的运行速度在所有计算机语言的程序中最快；D、开源，可以自由阅读源代码并对其进行改动
C、Python是一种解释型语言，通常比编译型语言（如C、C++）慢。

第二章 运算符、表达式与内置对象

填空题

- 表达式 `int(4**0.5)` 的值为2。
- Python内置函数 `len()` 可以返回列表、元组、字典、集合、字符串以及range对象中元素个数。
- 表达式 `'ab' in 'acbed'` 的值为False。
- 表达式 `sorted([111, 2, 33], key=lambda x: len(str(x)))` 的值为[2, 33, 111]。
- 表达式 `min(['11', '2', '3'])` 的值为'11'。
- 已知 `x=3` 和 `y=5` , 执行语句 `x, y = y, x` 后 `x` 的值为5。
- `print(1, 2, 3, sep='#')` 的输出结果为1#2#3。
- 已知列表对象 `x = ['11', '2', '3']` , 则表达式 `max(x, key=len)` 的值为'11'。
- 表达式 `3<5>2` 的值为True。
- 已知 `a = [1, 2, 3]` 和 `b = [1, 2, 4]` , 那么 `id(a[1])==id(b[1])` 的执行结果为True。
- 表达式 `[1, 2, 3]*3` 的执行结果为[1, 2, 3, 1, 2, 3, 1, 2, 3]。
- 表达式 `[3] in [1, 2, 3, 4]` 的值为False。
- 表达式 `{1, 2, 3} & {3, 4, 5}` 的执行结果为{3}。
- 表达式 `list(zip('abcd', [1, 2, 3]))` 的执行结果为[('a', 1), ('b', 2), ('c', 3)]。
- `print(0.1+0.2==0.3)` 的输出结果为False。

第三章 列表、元组、字典、集合与字符串

单选题

- 假设列表对象 `aList` 的值为 `[3, 4, 5, 6, 7, 9, 11, 13, 15, 17]` , 那么切片 `aList[3:7]` 得到的值是 (A)
A、[6, 7, 9, 11]; B、[6, 7, 9, 11,13]; C、[3, 4, 5, 6, 7]; D、[3, 4, 5, 6]
- 语句序列 `nums = set([1,2,2,3,3,3,4]); print(len(nums))`; 的运行结果是 (C)
A、1; B、2; C、4; D、7
- 语句 `list(range(0,8,2))` 执行结果为 (A)
A、[0,2,4,6]; B、[2,4,6]; C、[0,2,4]; D、[0,2,4,6,8]
- 已知字典 `data={'a': 97 , 'A': 65 }` , 那么 `data.get ('a', None)` 的值为 (A)
A、97; B、65; C、'a'; D、None
- 假设列表对象 `aList` 的值为 `[3, 4, 5, 6, 7, 9, 11, 13, 15, 17]` , 那么切片 `aList[2:6:2]` 得到的值是 (D)
A、[5, 6, 7, 9,11]; B、[5, 6,,7,9]; C、[5, 7,11]; D、[5,7]
- 长度为 `len` 的Python列表、元组和字符串中最后一个元素的下标为 (C)
A、len; B、len+1; C、-1; D、0
- 已知元组对象 `t=("cat", "dog", "tiger", "human")` , 则语句 `t[::-1]` 执行的结果是 (C)
A、{'human', 'tiger', 'dog', 'cat'}; B、['human', 'tiger', 'dog', 'cat']; C、('human', 'tiger', 'dog', 'cat'); D、运行出错
- 列表式 `[i*2 for i in 'Python']` 的执行结果是 (B)
A、Python Python; B、['PP', 'yy', 'tt', 'hh', 'oo', 'nn']; C、[2, 4, 6, 8, 10, 12]; D、错误
- 下面代码的输出结果是 (C)

```
1 | i = ['a', 'b', 'c']
2 | j = [1,2,3]
3 | b = dict(zip(i,j))
4 | print(b)
```

- A、('a': 1, 'b': 2, 'c': 3); B、(1: 'a', 2: 'd', 3: 'c'); C、{'a': 1, 'b': 2, 'c': 3}; D、{1: 'a', 2: 'd', 3: 'c'}
10. 下列代码执行的结果是 (B)

```
1 | for i in list(range(5)):
2 |     print(i,end=",")
```

- A、0,1,2,3,4; B、0,1,2,3,4; C、1,2,3,4,5; D、1,2,3,4,5,

填空题

- Python语句 `list(range(1,10,3))` 执行结果为[1, 4, 7]。
- 切片操作 `list(range(6))[::2]` 执行结果为[0, 2, 4]。
- 已知 `x = {1:2}` , 那么执行语句 `x[2] = 3` 之后, `x` 的值为{1: 2, 2: 3}。
- 表达式 `{1, 2, 3, 4} - {3, 4, 5, 6}` 的值为{1, 2}。

- 表达式 `set([1, 1, 2, 3])` 的值为{1, 2, 3}。
- 列表推导式 `[i for i in range(100) if i%13==0]` 得到的列表是[0, 13, 26, 39, 52, 65, 78, 91]。
- 已知 `x = [3, 5, 7]`，那么执行语句 `x[len(x):] = [1, 2]` 之后，`x` 的值为[3, 5, 7, 1, 2]。
- 表达式 `list(zip([1,2], [3,4]))` 的值为[(1, 3), (2, 4)]。
- 表达式 `[x for x in [1,2,3,4,5] if x<3]` 的值为[1, 2]。
- 表达式 `[index for index, value in enumerate([3,5,7,3,7]) if value == max([3,5,7,3,7])]` 的值为[2, 4]。
- `'#'.join('abcdefg'.split('cd'))` 的值为'ab#efg'。

对于 `'abcdefg'.split('cd')`，字符串 `'abcdefg'` 中包含子字符串 `'cd'`，因此按照 `'cd'` 分割后，返回的列表是 `['ab', 'efg']`；
`'#'.join(['ab', 'efg'])` 将列表中的元素以 `'#'` 为分隔符连接起来，得到字符串 `'ab#efg'`。

- 已知字符串 `x = 'hello world'`，那么执行语句 `x.replace('hello', 'hi')` 之后，`x` 的值为'hello world'。

字符串是不可变的。

- 表达式 `'ab' in 'acbed'` 的执行结果为False。
- 已知 `x = 'abcdef'`，则表达式 `x[3:] + x[:3]` 的值为'defabc'。
- `"mysqlsqlserverPostgresQL".count("sql", 3)` 输出的结果是1。

从字符串的第4个字符开始查找'sql'字符串出现的次数。

第四章 正则表达式

填空题

- 假设 `re` 模块已导入，那么表达式 `re.findall('(\\d)\\1+', '33abcd112')` 的值为['3', '1']。
`\\d` 表示匹配一个数字字符，并将其存储为一个组；`\\1+` 表示匹配一个或多个前面捕获的数字；所以 `(\\d)\\1+` 会匹配连续的、相同的数字序列。
- `print(re.match('abc', 'deabcfg'))` 的运行结果是None。
`re.match` 从字符串的开头开始匹配。如果字符串的开头不符合正则表达式，则匹配失败。
- 代码 `print(re.match('^a-zA-Z]+$','abcDEFG000'))` 的输出结果为None。
`^` 表示字符串的开头；`[a-zA-Z]+` 表示一个或多个字母，大小写都可以；`$` 表示字符串的结尾；因此，正则表达式要求整个字符串只能由字母组成。
- 假设正则表达式模块 `re` 已导入，那么表达式 `re.sub('\\d+', '*', 'a12345bb67c890d0e')` 的值为'a*bb*c*d*e'。
`\\d+` 表示匹配一个或多个数字；`*` 是替换的内容；`re.sub` 会将所有匹配的数字替换为 `*`。
- 已知字符串 `x = 'hello world'`，那么执行语句 `x.replace('hello', 'hi')` 之后，`x` 的值为'hello world'。
字符串不可变，要将替换后的字符串赋值给新变量才行。

第五章 程序控制结构，函数

选择题

- 以下代码输出结果为（B）

```
1 def printLine(text):
2     print(text , ' Runoob')
3 printLine('Python')
```

A、Python, Runboob; B、Python Runboob; C、text Runoob; D、Runoob

- 如果函数没有使用 `return` 语句，则函数返回的是（B）
A、0; B、None对象; C、任意整数; D、错误!函数必须要有返回值
- 以下代码输出的结果是（B）

```
1 numbers = [1,3,6]
2 newNumbers = tuple(map(lambda x:x,numbers))
3 print(newNumbers)
```

A、[1,3,6]; B、(1,3,6); C、[2,6,12]; D、(2,6,12)

- 以下代码输出的结果是（C）

```
1 def demo(a, b, c=5):
2     print(a, b, c)
3 demo(c=8, a=9, b=0)
```

A、8 9 0; B、8 9 5; C、9 0 8; D、9 0 5

- 以下代码输出的结果是（A）

```
1 def demo(a, b, c):
2     print(a+b+c)
3 dic = {1:'a', 2:'b', 3:'c'}
4 demo(*dic)
5 demo(*dic.values())
```

A、6 abc; B、abc 6; C、6 6; D、abc abc

6. 以下代码输出的结果是 (A)

```
1 def demo(a, b, c):
2     print(a+b+c)
3 dic = {'a':3, 'b':4, 'c':5}
4 demo(**dic)
```

A、12; B、abc; C、'a':3,'b':4,'c':5; D、None

7. 以下代码输出的结果是 (C)

```
1 L = [1,2,3,4,5]
2 print(list(map(lambda x: x+6, L)))
```

A、[1, 2, 3, 4, 5,6]; B、1, 2, 3, 4, 5,6; C、[7, 8, 9, 10, 11]; D、7, 8, 9, 10, 11

8. 以下代码输出的结果是 (C)

```
1 def func(a,b):
2     c=a**2+b
3     b=a
4     return c
5 a=10
6 b=100
7 print(func(a,b)+a)
```

A、120; B、130; C、210; D、300

9. 以下代码输出的结果是 (D)

```
1 for s in "HelloWorld":
2     if s=="w":
3         continue
4     print(s,end="")
```

A、World; B、Hello; C、HelloWorld; D、Helloorld

10. 对以下代码描述错误的是 (B)

```
1 import random
2 num = random.randint(1,10)
3 while True:
4     guess = input()
5     i = int(guess)
6     if i == num:
7         print("你猜对了")
8         break
9     elif i < num:
10        print("小了")
11    elif i > num:
12        print("大了")
```

A、random.randint(1,10)生成一个[1,10]之间的随机整数; B、“import random”这行代码是可以省略的; C、这段代码实现了简单的猜数字游戏; D、“while True:”创建了一个无条件执行的无限循环

填空题

1. 下列程序输出的结果是7 5 3 2。

```
1 for n in range(10, 1, -1):
2     for i in range(2, n):
3         if n%i == 0:
4             break
5     else:
6         print(n, end=' ')
```

2. 下列循环中的print调用多少次? 45

```
1 for i in range(10):
2     for j in range(i):
3         print(i * j)
```

3. 下列代码中循环体会执行多少次? 7

```
1 number = 20
2 while number > 0:
3     number -= 3
4     print(number, end = ' ')
```

4. 下列程序输出的结果是3,6,9,12,18,

```
1 for num in range(0, 20, 3):
2     if num%5 != 0:
3         print(num, end=' , ')
```

5. 下列程序输出的结果是{'优秀': 2, '良': 4, '中': 5, '及格': 1, '不及格': 2}

```
1 scores = [89,70,49,87,92,63,71,78,81,79,82,75,90,54]
2 groups = {'优秀':0, '良':0, '中':0, '及格':0, '不及格':0}
3 for score in scores:
4     if score>=90:
5         groups['优秀'] = groups['优秀']+1
6     elif score>=80:
7         groups['良'] = groups['良']+1
8     elif score>=70:
9         groups['中'] = groups['中']+1
10    elif score>=60:
11        groups['及格'] = groups['及格']+1
12    else:
13        groups['不及格'] = groups['不及格']+1
14    print(groups)
```

6. 执行以下程序, 如果输入"93python22", 则输出结果是93python22

```
1 w = input('请输入数字和字母构成的字符串:')
2 for x in w:
3     if '0'<= x <= '9':
4         continue
5     else:
6         w.replace(x, '')
7 print(w)
```

7. 以下程序的输出结果是ababab

```
1 for i in range(3):
2     for s in "abcd":
3         if s=="c":
4             break
5         print (s,end="")
```

8. 下列程序共输出12个值

```
1 age = 23
2 start = 2
3 if age % 2 != 0:
4     start = 1
5 for x in range(start, age + 2, 2):
6     print(x)
```

9. 执行以下程序, 如果分别输入'p'、'q', 输出结果是3

```
1 k = 0
2 while True:
3     s = input()
4     if s != 'q':
5         k += 1
6         continue
7     else:
8         k += 2
9         break
10    print(k)
```

10. 下列程序输出的结果是7

```
1 for n in range(10, 1, -1):
2     for i in range(2, n):
3         if n % i == 0:
4             break
5     else:
6         print(n, end='')
7         break
```

第六章 文件操作

选择题

1. 以下选项中，不是Python对文件的读操作方法的是（D）

A、read；B、readline；C、readlines；D、readtext

A、`read()`：读取整个文件的内容；B、`readline()`：读取文件的一行；C、`readlines()`：读取文件的所有行，并将它们作为一个列表返回。

2. 对以下代码描述错误的是（A）

```
1 fname = input("请输入要打开的文件：")
2 fi = open(fname, "r")
3 for line in fi.readlines():
4     print(line)
5 fi.close()
```

A、通过fi.readlines()方法将文件的全部内容读入一个字典fi；B、实现功能：用户输入文件路径，以文本文件方式读入文件内容并逐行打印；C、通过fi.readlines()方法将文件的全部内容读入一个列表fi；D、上述代码中fi.readlines()可以优化为fi

3. 对以下代码描述错误的是（A）

```
1 fname = input("请输入要写入的文件：")
2 fo = open(fname, "w+")
3 ls = ["清明时节雨纷纷，", "路上行人欲断魂，", "借问酒家何处有？", "牧童遥指杏花村。"]
4 fo.writelines(ls)
5 fo.seek(0)
6 for line in fo:
7     print(line)
8 fo.close()
```

A、fo.seek(0)这行代码可以省略，不影响输出效果；B、fo.writelines(ls)将元素全为字符串的ls列表写入文件；C、代码主要功能为向文件写入一个列表类型，并打印输出结果；D、执行代码时，从键盘输入“清明.txt”，则清明.txt被创建

A、在写入文件后，文件的指针位于文件的末尾。如果不将文件指针重置到文件的开头（通过 `fo.seek(0)`），在迭代文件对象 `fo` 时将不会读取任何内容，因此不会打印出任何东西。

4. 以下选项中，不是Python文件打开的合法模式组合是（B）

A、"a+"; B、"t+"; C、"r+"; D、"w+"

- `r`：以只读方式打开文件，如果文件不存在则抛出异常。
- `w`：以写入方式打开文件，如果文件存在则覆盖，不存在则创建。
- `a`：以追加方式打开文件，如果文件存在则在文件末尾追加内容，不存在则创建。
- `b`：以二进制模式打开文件。
- `+`：表示文件打开后可以读写。
- `r+`：读写模式。
- `w+`：读写模式，如果文件存在则覆盖。
- `a+`：读写模式，如果文件存在则在文件末尾追加内容。

5. 当打开一个不存在的文件时，以下选项中描述正确的是（C）

A、文件不存在则创建文件；B、一定会报错；C、根据打开类型不同，可能不报错；D、不存在文件无法被打开

只有在写入模式（`w`）或追加模式（`a`）时，如果文件不存在，Python 才会创建文件。

6. 关于下面代码中的变量x，以下选项中描述正确的是（C）

```
1 fo = open(fname, "r")
2 for x in fo:
3     print(x)
4 fo.close()
```

A、变量x表示文件中的一个字符；B、变量x表示文件中的全体字符；C、变量x表示文件中的一行字符；D、变量x表示文件中的一组字符

7. 下列选项描述错误的是（B）

- A、JSON 是一种流行的文件格式，主要用于在web应用程序中存储和传输数据；B、JSON模块是Python的标准库，使用前不需要用import json语句导入；C、JSON模块是扩展库，使用前需要使用pip工具安装；D、json.load()用于从文件中读取json数据，json.dump()用于将数据写入到json文件
8. 关于CSV文件的描述，以下选项中错误的是（D）
- A、CSV文件的每一行是一维数据，可以使用Python中的列表类型表示；B、整个CSV文件是一个二维数据；C、CSV文件格式是一种通用的、相对简单的文件格式，应用于程序之间转移表格数据；D、CSV文件通过多种编码表示字符
- == CSV 文件的编码通常是UTF-8、ASCII或其他字符编码，但是统一编码的。==
9. 对于特别大的数据文件，以下选项中描述正确的是（C）
- A、选择内存大的计算机，一次性读入再进行操作；B、Python可以处理特别大的文件，不用特别关心；C、使用for .. in..循环，分行读入，逐行处理；D、Python无法处理特别大的数据文件
10. 关于上下文管理语句with的描述错误的是（D）
- A、关键字with可以自动管理资源，能确保文件被正确关闭；B、读写文件应该优先考虑使用with语句；C、一个with语句中可以同时建立两个文件操作对象；D、with关键字只能用于文件打开操作，不能实现数据库连接或网络连接
- with 语句在 Python 中被称为上下文管理器，它不仅可以用于文件打开操作，还可以用于管理任何需要设置和清理的资源，包括但不限于数据库连接、网络连接、线程锁等。with 语句的主要优点是它能够在代码块执行完毕后自动调用资源的清理方法（如文件的close 方法），即使在代码块中发生异常也是如此。这是通过上下文管理器的__enter__ 和 __exit__ 方法实现的。

第七章 numpy数组与矩阵运算

选择题

1. 以下关于NumPy数组与列表的说法，错误的是（D）
- A、数组和列表都支持切片访问；B、列表可以转换为NumPy数组；C、数组支持布尔型索引访问，列表不支持；D、数组和列表都可以有多个维度
2. np是numpy模块的别名。若 `a=np.array((2,4,6))`，则 `a**2` 完成的操作是（C）
- A、数组中的每个元素乘以2；B、数组中的第一个元素乘以2；C、计算数组中每个元素的2次方；D、用数组中的每个元素作为2的幂次方
3. 以下关于数组对象的reshape方法的描述中，错误的是（A）
- A、reshape方法可以改变数组对象的大小；B、reshape方法可以将一维数组变换为二维数组；C、reshape方法可以将二维数组变换为一维数组；D、执行reshape方法后会返回一个新数组
4. a为NumPy数组对象，执行 `a.shape` 的结果为(5,)，以下说法正确的是（B）
- A、a的维度为5，a的大小为1；B、a的维度为1，a的大小为5；C、a的维度为2，a的大小为5；D、a的维度和大小都是5
5. np是numpy模块的别名。若 `a= np.array([[11,12,13,14],[15,16,17,18]])`，则 `a[a>16]` 的结果是（D）
- A、True；B、False；C、[17,18]；D、array([17,18])
6. np是numpy模块的别名。若 `a=np.array([[11,12,13, 14],[15,16,17,18]])`，则 `a[1,1:3]` 的结果是（D）
- A、array([11, 12, 13])；B、array([15, 16, 17])；C、array([11, 13])；D、array([16, 17])
7. 以下关于NumPy数组的说法，正确的是（B）
- A、数组中只能存储数字，不能存储字符串；B、数组的形状是可以变换的；C、数组的每个轴的长度是不能改变的；D、数组的维度数表示数组元素的数量
8. np是numpy模块的别名。执行 `np.arange(5)`，返回的结果是（D）
- A、[1,2,3,4]；B、[1,2,3,4,5]；C、array([0, 1, 2, 3, 4, 5])；D、array([0, 1, 2, 3, 4])

填空题

1. 下述代码的输出结果为[0 7 0 27 13 0 71]

```
1 import numpy as np
2 the_array = np.array([49, 7, 44, 27, 13, 35, 71])
3 an_array = np.where((the_array > 30) & (the_array < 50), 0, the_array)
4 print(an_array)
```

2. 下述代码的输出结果为[1 0 1 1 0 1 1]

```
1 import numpy as np
2 the_array = np.array([49, 7, 44, 27, 13, 35, 71])
3 an_array = np.asarray([0 if val < 25 else 1 for val in the_array])
4 print(an_array)
```

3. 下述代码的输出结果为

```
[[0 1 0 1]
 [1 0 1 0]
 [0 1 0 1]
 [1 0 1 0]]
```

```
1 data = np.zeros((4, 4), dtype=int)
2 data[1::2, ::2] = 1
3 data[:, 1::2] = 1
4 print(data)
```

4. 下述代码的输出结果为

[[1 2]
[5 6]]

```
1 import numpy as np
2 the_arr = np.array([[0, 1, 2, 3, 5, 6, 7, 8],
3                     [4, 5, 6, 7, 5, 3, 2, 5],
4                     [8, 9, 10, 11, 4, 5, 3, 5]])
5 print(the_arr[0:2,1:3])
```

5. 下述代码的输出结果为[6 15 24 33]

```
1 import numpy as np
2 arr = np.arange(1, 13)
3 newarr = arr.reshape(4, 3)
4 column_sums = newarr.sum(axis=1)
5 print(column_sums)
```

6. 下述代码的输出结果为48

```
1 import numpy as np
2 arr = np.arange(1, 13)
3 newarr = arr.reshape(4, 3)
4 column_sums = newarr[:, 0:2].sum()
5 print(column_sums)
```

第八章 pandas数据分析

选择题

1. 查看DataFrame对象中是否存在重复值的方法是（C）
A、info(); B、copy(); C、duplicated(); D、repeat()
A、此方法用于打印关于DataFrame的简要信息，包括索引类型、列数、非空值的数量等，但不会直接显示重复值；B、此方法用于返回DataFrame的一个副本；D、此方法用于沿着某个轴重复DataFrame的元素。
2. Pandas中两个主要的数据结构是（C）
A、List、Tuple； B、Set、Dict； C、Series、DataFrame； D、Table、ndarray
3. 下列语句可以用来查看 df 前五行的是（C）
A、df.tail(5); B、df.describe(); C、df.head(); D、df.index()
A、此方法默认显示DataFrame的最后五行；B、此方法用于生成描述性统计；D、此方法返回DataFrame的索引值。
4. 下列语句可以用来选取df中列名为A和B的两列的是（C）
A、df['A','B']; B、df(A,B); C、df[['A','B']]; D、df(['A','B'])
5. 下列哪个语句可以对df的前三行进行切片（B）
A、df[1:4]; B、df[0:3]; C、df([0:3]); D、df([1:4])
6. 对于语句 df = pd.DataFrame(np.random.randn(5,2), index=range(2,7), columns=list('AB'))，下列哪个语句不能正确获取df中的第三行数据的选项是（A）
A、df.iloc[3]; B、df.loc[[4]]; C、df.iloc[[2]]; D、df.loc[4]

```
1      A      B
2  2  -0.027044  0.318311
3  3  -0.388954 -0.542876
4  4   1.136310  0.721301
5  5   0.027524 -0.720942
6  6   0.374353  0.145009
7  ---
8  iloc是位置索引，loc是标签索引
```

7. 以下语句可以用来选取df中“A”列取值大于0 的行的是（A）
A、df[df.A > 0]; B、df['A' > 0]; C、df['A'] > 0; D、df[df.'A' > 0]
8. 删除df中所有包含空值的列可以采用（D）
A、df.drop(); B、df.dropna(); C、df.dropna(axis=0); D、df.dropna(axis=1)
9. 用‘空’替换dataframe对象中所有的空值可以用以下哪个语句（A）
A、df.fillna(“空”); B、df.isnull(“空”); C、df.tail(“空”); D、df.replace(“空”)

10. 下述代码的输出结果为（D）

```
1 | import pandas as pd
2 | Data=pd.DataFrame([[2,3],]*3, columns=['A', 'B'])
3 | B=Data.apply(lambda x:x+1)
4 | print (B.loc[1, 'B'])
5 | ---
6 | Data:
7 |      A  B
8 | 0    2  3
9 | 1    2  3
10 | 2    2  3
11 | ---
12 | B:
13 |      A  B
14 | 0    3  4
15 | 1    3  4
16 | 2    3  4
```

A、 3； B、 1； C、 2； D、 4

第九章 matplotlib数据可视化

选择题

1. 使用matplotlib.pyplot中的（A）函数可以绘制折线图
A、 plot； B、 bar； C、 hist； D、 pie
2. 使用matplotlib绘图时，设置图例的函数是（B）
A、 title； B、 legend； C、 xlabel； D、 text
3. 绘制既可以展示数据分布，也可以展示数据之间相关性的图表，使用的函数是（D）
A、 pie； B、 bar； C、 plot； D、 scatter
4. 使用Pandas中的方法绘图时，图表的数据来源是（B）
A、 Numpy数组对象中的数据； B、 Series或DataFrame对象中的数据； C、 列表中的数据； D、 以上都可以
5. 使用matplotlib.pyplot中的（B）函数可以绘制柱形图
A、 plot； B、 bar； C、 scatter； D、 pie
6. matplotlib.pyplot中设置图形标题的函数是（B）
A、 xlabel； B、 title； C、 head； D、 name
7. 比较适合描述和比较多组数据随时间变化的趋势，或者一组数据对另外一组数据的依赖程度的图表是（A）
A、 折线图； B、 散点图； C、 饼状图； D、 雷达图
8. 要绘制图形直观地表现企业中各个团队的绩效，评估团队在多个维度上的表现，例如销售额、市场份额、客户满意度等，最适合的绘图函数是（D）
A、 pie； B、 plot； C、 scatter； D、 polar
9. 在matplotlib子图绘制中,若执行plt.subplot(3,2,4),则当前的绘图子区域索引号是（C）
A、 2； B、 3； C、 4； D、 6
10. matplotlib.pyplot中的函数text()的作用是（D）
A、 给坐标系增加标题； B、 给坐标轴增加文本标签； C、 给坐标轴增加题注； D、 在坐标系任意位置增加文本