

第二章作业

7. 简答题

根据列表list1=[1,2,3,4,3,1]，使用if-elif-else命令，当判断结果为1时，输出"你好"，当判断结果为3时，输出"欢迎"，当判断结果为其他数值时，输出"再见"。

```
In [1]: list1 = [1, 2, 3, 4, 3, 1]
```

```
for item in list1:
    if item == 1:
        print('你好')
    elif item == 3:
        print('欢迎')
    else:
        print('再见')
```

你好
再见
欢迎
再见
欢迎
你好

8. 简答题

使用 for 函数对 1~10的整数进行求和，并输出求和结果。

```
In [2]: total = 0
for i in range(1, 11):
    total += i
print('1~10的整数求和结果为:', total)
```

1~10的整数求和结果为: 55

9. 简答题

根据字典 a={'a': 2,'b': 1,'c':3,'d':4}，输出每一个键所对应的值。

```
In [3]: a = {'a': 2, 'b': 1, 'c': 3, 'd': 4}

for key in a:
    print(f'键 {key} 对应的值为: {a[key]}')
```

键 a 对应的值为: 2
键 b 对应的值为: 1
键 c 对应的值为: 3
键 d 对应的值为: 4

10. 操作题

第二章Numpy补充章节任务

1、用numpy定义一个4行5列的随机数组，并输出

```
In [4]: import numpy as np

arr = np.random.rand(4, 5)
print('4行5列的随机数组:')
print(arr)
```

4行5列的随机数组:

```
[[0.74679525 0.3417908  0.17058653 0.43121576 0.03757114]
 [0.17268403 0.1303191  0.90600836 0.96461496 0.04184161]
 [0.79703604 0.69756071 0.5044036  0.07695862 0.71116542]
 [0.52174742 0.2961738  0.61513951 0.86648819 0.28095563]]
```

2、先使用range创建一个1~12的整数list列表，并将其转化为numpy数组

```
In [5]: list2 = list(range(1, 13))
print('list列表:', list2)

arr2 = np.array(list2)
print('numpy数组:', arr2)
```

list列表: [1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12]

numpy数组: [1 2 3 4 5 6 7 8 9 10 11 12]

3、将上述（2）中的一维数组转化为3行4列数据的形状，并输出数据维度、形状、数据类型、数组元素

```
In [6]: arr3 = arr2.reshape(3, 4)
print('3行4列数组:')
print(arr3)
print('维度:', arr3.ndim)
print('形状:', arr3.shape)
print('数据类型:', arr3.dtype)
print('数组元素:')
for row in arr3:
    for elem in row:
        print(elem, end=' ')
    print()
```

3行4列数组:

```
[[ 1  2  3  4]
 [ 5  6  7  8]
 [ 9 10 11 12]]
```

维度: 2

形状: (3, 4)

数据类型: int64

数组元素:

```
1 2 3 4
5 6 7 8
9 10 11 12
```

4、生成一个0-30的等差数列 跨度为3，并输出数据结构的形状，数据的列表名，数据的索引，数据的类型

```
In [7]: import pandas as pd

arr4 = np.arange(0, 31, 3)
print('等差数列:', arr4)

s = pd.Series(arr4)
print('形状:', s.shape)
print('列表名(name):', s.name)
print('索引:', s.index.tolist())
print('数据类型:', s.dtype)
```

等差数列: [0 3 6 9 12 15 18 21 24 27 30]

形状: (11,)

列表名(name): None

索引: [0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10]

数据类型: int64

5、输出两个相同维度的数组。并执行行和输出相加，相减，相乘

```
In [8]: a1 = np.array([[1, 2, 3], [4, 5, 6]])
a2 = np.array([[7, 8, 9], [10, 11, 12]])
print('数组a1:')
print(a1)
print('数组a2:')
print(a2)
print('\n相加:')
print(a1 + a2)
print('\n相减:')
print(a1 - a2)
print('\n相乘:')
print(a1 * a2)
```

```
数组a1:  
[[1 2 3]  
 [4 5 6]]  
数组a2:  
[[ 7  8  9]  
 [10 11 12]]
```

```
相加:  
[[ 8 10 12]  
 [14 16 18]]
```

```
相减:  
[[-6 -6 -6]  
 [-6 -6 -6]]
```

```
相乘:  
[[ 7 16 27]  
 [40 55 72]]
```

6、输出一维、二维数组的最小元素，最大元素，中位数，平均数，标准方差

```
In [9]: arr_1d = np.array([3, 1, 4, 1, 5, 9, 2, 6])  
arr_2d = np.array([[1, 2, 3], [4, 5, 6], [7, 8, 9]])  
  
print('一维数组:', arr_1d)  
print('最小元素:', arr_1d.min())  
print('最大元素:', arr_1d.max())  
print('中位数:', np.median(arr_1d))  
print('平均数:', arr_1d.mean())  
print('标准方差:', arr_1d.std())  
  
print('\n二维数组:')  
print(arr_2d)  
print('最小元素:', arr_2d.min())  
print('最大元素:', arr_2d.max())  
print('中位数:', np.median(arr_2d))  
print('平均数:', arr_2d.mean())  
print('标准方差:', arr_2d.std())
```

一维数组: [3 1 4 1 5 9 2 6]
最小元素: 1
最大元素: 9
中位数: 3.5
平均数: 3.875
标准方差: 2.5708704751503917

二维数组:
[[1 2 3]
 [4 5 6]
 [7 8 9]]
最小元素: 1
最大元素: 9
中位数: 5.0
平均数: 5.0
标准方差: 2.581988897471611

7、定义一个范围为3~8的数组，输出一个大于4并且除2为0的数

```
In [10]: arr7 = np.arange(3, 9)
print('数组:', arr7)
result = arr7[(arr7 > 4) & (arr7 % 2 == 0)]
print('大于4并且除2为0的数:', result)
```

数组: [3 4 5 6 7 8]
大于4并且除2为0的数: [6 8]

8、完成两个数组的

a=np.arange(3,8), b=np.arange(6,9)的交集并输出

```
In [11]: a = np.arange(3, 8)
b = np.arange(6, 9)
print('数组a:', a)
print('数组b:', b)
print('交集:', np.intersect1d(a, b))
```

数组a: [3 4 5 6 7]
数组b: [6 7 8]
交集: [6 7]

9、生成今天、明天、昨天的日期，包含年月日

```
In [12]: today = np.datetime64('today')
yesterday = today - np.timedelta64(1, 'D')
tomorrow = today + np.timedelta64(1, 'D')

print('昨天:', yesterday)
print('今天:', today)
print('明天:', tomorrow)
```

昨天: 2026-04-13
今天: 2026-04-14
明天: 2026-04-15

