

人工智能编程基础纳米学位

项目知识点练习册

项目：（P1 机器人走迷宫预备练习题）

在这个notebook文件中，我们设计了多项练习题来帮助你为之后的实战项目作准备，并希望通过编程练习来帮助你更好地掌握前几周内所学习的内容！

有些模板代码已经提供给你，但你还需要实现更多的功能来完成这个练习。除非有明确要求，你无须修改任何已给出的代码。每一个练习都配有详细的指导，需要实现的部分也会在注释中以'TODO'标出。

除了实现代码外，你还需要回答一些与项目及代码相关的问题。每个需要回答的问题都会以 '**问题 X**' 标记。请仔细阅读每个问题，并且在问题后的 '**回答**' 部分写出完整的答案。助教将根据 你对问题的回答 和 撰写代码实现的功能 来对你提交的练习进行评分。

提示： Code 和 Markdown 区域可通过 **Shift + Enter** 快捷键运行。此外，Markdown可以通过双击进入编辑模式。

让我们开始吧！

Task 0：数据类型和运算

问答题

问题：

python有几种基本数据类型，具体是哪些？这些数据类型中哪些是可变的，哪些是不可变的？哪些是有序的，哪些是无序的？

回答:

基本数据类型	是否可变	是否有序	是否可包含重复数据
Number (数字)	否		
String (字符串)	否	是	是
Tuple (元组)	否	是	是
List (列表)	是	是	是
Set (集合)	是	否	否
Dictionary (字典)	是	否	否

可变数据类型的特点:

定义一个变量 v, 将其传入一个可调用对象 (函数、lambda表达式、函数类) a_function(v_input), 在a_function中改变v_input中的值, 若type(v)是可变类型, 则v的值随之改变, 若type(v)是不可变类型, 则v的值保持不变。示例代码:

```
def a_function(v_input):
    if type(v_input) == list:
        v_input.append(2019)
    elif type(v_input) == str:
        v_input = 'hello 2019'

v1 = [1, 2, 3]
v2 = 'bye 2018'

print('before call a_function:')
print('v1=', v1)
print('v2=', v2)

a_function(v1)
a_function(v2)

print('after call a_function:')
print('v1=', v1)
print('v2=', v2)
```

In [1]:

```
def a_function(v_input):
    if type(v_input) == list:
        v_input.append(2019)
    elif type(v_input) == str:
        v_input = 'hello 2019'

v1 = [1, 2, 3]
v2 = 'bye 2018'

print('before call a_function:')
print('v1=', v1)
print('v2=', v2)

a_function(v1)
a_function(v2)

print('after call a_function:')
print('v1=', v1)
print('v2=', v2)
```

```
before call a_function:
v1= [1, 2, 3]
v2= bye 2018
after call a_function:
v1= [1, 2, 3, 2019]
v2= bye 2018
```

任务描述

熟悉基本数据类型。

练习

练习0

新建一个二维列表，列表名定义为matrix，列表的内容如下：

第一项：0,1,2

第二项：3,4,5

第三项：6,7,8

注意，二维列表可以当做矩阵来看，某一项为行，某一项中的第几个为列，比如数字1处于matrix列表的第一行第二列，用索引值来看的话就是行0，列1（因为所以是从零开始计数的）。以下都用行列来表示了。

In [22]:

```
### TODO: 完成这个列表的创建
matrix = [list(range(0, 3)), list(range(3, 6)), list(range(6, 9))]
```

In [23]:

```
matrix
```

Out[23]:

```
[[0, 1, 2], [3, 4, 5], [6, 7, 8]]
```

练习1

使用列表方法给matrix列表新增一项（行），项（行）的内容为 'bj','gz','sh'

In [28]:

```
### TODO: 使用字符串方法给matrix列表添加
matrix.append(['bj', 'gz', 'sh'])
matrix
```

Out[28]:

```
[[0, 1, 2], [3, 4, 5], [6, 7, 8], ['bj', 'gz', 'sh']]
```

练习2

- 1、用代码输出matrix第二行第三列的值（应该为5）
- 2、用代码输出matrix第四行的值（应该输出bj,gz,sh）
- 3、用代码输出matrix的行数和列数(应该输出4和3)

In [7]:

```
matrix_2_3 = matrix[1][2]
print(matrix_2_3)
matrix_row4 = matrix[3]
print(matrix_row4)
rows = len(matrix)
cols = len(matrix[0])
print(rows, cols)
```

```
5
['bj', 'gz', 'sh']
4 3
```

练习3

新建一个字典，字典的名字为loc，将新增的项（'bj','gz','sh'）分别作为字典的键（key），bj的值（value）为matrix的第一行的和（输出时能看到值为3），gz的值为matrix的第二行的和（输出时能看到值为12），sh的值为matrix的第三行（输出时能看到值为21）。

In [24]:

```
### TODO: 新建字典loc, 新建时可以使用之前习题所使用的方法, 不要重新输入具体的值来创建
loc = {}
for i, k in enumerate(('bj', 'gz', 'sh')):
    loc[k] = sum(matrix[i])

loc
```

Out[24]:

```
{'bj': 3, 'gz': 12, 'sh': 21}
```

练习4

输出字典loc中键sh的值是多少

In [25]:

```
### TODO: 已知字典key求该key的value值
loc['sh']
```

Out[25]:

```
21
```

In [26]:

```
loc.get('sh')
```

Out[26]:

```
21
```

练习5

字符串连接, 连接matrix第四行的所有元素并输出

In [31]:

```
### TODO:
print('=='.join(matrix[3]))
```

```
bj==gz==sh
```

Task 1: 控制流

问答题

问题 0:

if判断和while后面跟的是什么，或者说什么情况下会进入判断体和循环体（判断循环缩进下的内容）？

答: if判断和while后面跟的都是**执行条件**

一个空列表作为if判断或者是while后面跟的东西的话，是否会进入判断体或者循环体（判断循环缩进下的内容）？

回答：一个空列表作为if判断或者是while后面跟的东西的话，不会进入判断体或者循环体。此外，用None, 0, "", {}, (), [],作为执行条件的话，也不会进入循环。

问题 1:

break和continue的区别？

回答：

break是跳出循环体，不再执行该循环体的任何循环；continue掠过本次循环，继续下次循环。

```
for i in range(6):
    if i == 2:
        break
    print(i)
```

```
for i in range(6):
    if i == 2:
        continue
    print(i)
```

In [8]:

```
for i in range(6):
    if i == 2:
        break
    print(i)
```

0
1

In [9]:

```
for i in range(6):
    if i == 2:
        continue
    print(i)
```

0
1
3
4
5

问题 2:

请简要描述迭代器，生成器和可迭代对象的关系。

回答: <https://zhuanlan.zhihu.com/p/36810728> (<https://zhuanlan.zhihu.com/p/36810728>)

循环能做什么

循环能让一段重复逻辑的工作自动执行，并在一定条件时候开始，一定条件时候结束。我们可以利用循环每隔一段时间去检查设备的温度是否正常，检查网络链接是否正常；还可以利用循环处理大规模数据，筛选出相同条件的数据，近乎每个程序中都离不开循环。

为什么要用循环？

程序处理的数据一般是数以千计的，加入我们有一个列表students，存了1000名学生信息，包括姓名、性别、考试成绩等。现在要求我们找出，在本次考试中，成绩在60~80之间的学生名单。那么，如果不用循环，我们只能一行一句代码去判断学生的成绩是否满足条件，我们需要写至少1000行。但是如果用循环，我们只需要写一个循环题，不超过3行的代码既可以完成。极大的提高了开发效率，更重要的是代码维护的难度。

循环需要注意的地方

```
data = ['a', 'b', 'c']

# i 是索引, data[i]是data中的元素
for i in range(len(data)):
    print('第{}个元素是{}'.format(i, data[i]))

# e 是data中的元素
i = 0
for e in data:
    print('第{}个元素是{}'.format(i, e))
    i += 1

# i是索引, e 是data中的元素
for i, e in enumerate(data):
    print('第{}个元素是{}'.format(i, e))
```

In [10]:

```
data = ['a', 'b', 'c']

# i 是索引, data[i]是data中的元素
for i in range(len(data)):
    print('第{}个元素是{}'.format(i, data[i]))

# e 是data中的元素
i = 0
for e in data:
    print('第{}个元素是{}'.format(i, e))
    i += 1

# i是索引, e 是data中的元素
for i, e in enumerate(data):
    print('第{}个元素是{}'.format(i, e))
```

第0个元素是a
第1个元素是b
第2个元素是c
第0个元素是a
第1个元素是b
第2个元素是c
第0个元素是a
第1个元素是b
第2个元素是c

任务描述

针对控制流的练习。

练习

练习0

循环输出列表matrix的每一个值，每一行。

In [14]:

```
### TODO: 循环
for i in range(0, rows, 2):
    for j in range(len(matrix[i])):
        print(matrix[i][j])
```

0
1
2
6
7
8

练习1

找出数字7所在的索引，输出行列索引，另外要求找到后必须立即停止循环。

In [21]:

```
### TODO: 循环判断
count = 0
for x in range(rows)[:3]:
    for y in range(cols):
        if matrix[x][y] % 2 == 0:
            count += 1
count
```

Out[21]:

5

练习2

请写一个列表表达式，将matrix中的数据全部提取出来变成一个一维列表merge，最后输出该列表

In [17]:

```
merge = [j for i in matrix for j in i]
print(merge)
```

[0, 1, 2, 3, 4, 5, 6, 7, 8, 'bj', 'gz', 'sh']

Task 2: 函数

问答题

问题 0:

解释下return的作用，return了什么？return的东西去了哪里？

回答:

return就是为了告诉函数，函数的调用者能通过这个函数得到什么。

`a = sum([1, 2, 3])`，`sum()` 函数return的结果去了赋值符合= 的左边，传递给了 `a` 。如果返回结果没有传递，则返回值被丢弃。

问题 1:

代码1、

```
a = 100
def test_score(a):
    print(a)
test_score(95)
```

代码2、

```
a = 100
def test_score(a = 90):
    print(a)
test_score()
```

代码3、

```
a = 100
def test_score():
    print(a)
test_score()
```

这三个代码分别输出什么？同样输出a为什么造成这样的结果，请给予解释造成这个输出结果的原因。

In [36]:

```
a = 1000
def test_score(a):
    print(a)

test_score(a=95)
```

95

In [39]:

```
#####
def test_score(a = 90):
    print(a)
#####
a = 100
test_score(a = a)
```

100

In [41]:

```
a = 1000
def test_score():
    print(a)
test_score()
```

1000

回答:

问题 2: 代码1、

```
def test_score(a):  
    a = 95  
a = 100  
test_score(a)  
print(a)
```

代码2、

```
def test_score(a):  
    a.append(95)  
a = [100]  
test_score(a)  
print(a)
```

这两个代码分别输出什么？同样传入参数a为什么造成这样的结果，请给予解释造成这个输出结果的原因。

In [42]:

```
def test_score(a):  
    a = 95  
a = 100  
test_score(a)  
print(a)
```

100

In [43]:

```
def test_score(a):  
    a.append(95)  
a = [100]  
test_score(a)  
print(a)
```

[100, 95]

回答：

问题 3：

定义函数的时候，函数的参数的名字，是否必须跟调用时传入的参数名字相同？大概解释下你的答案。

****回答**：**

问题 4：

请解释下缩进在python中有什么作用。

回答：

任务描述

主要是对函数部分的掌握

练习

练习0

请写一个函数,函数名: get_index, 函数有一个默认参数value, 默认值为0, 还有一个必须参数mat (是一个二维列表, 请务必保证上面练习题中的matrix正确生成并运行过, 测试时调用函数要用), 该函数的功能是返回value所对应mat上的索引 (不需要考虑value不存在于matrix的情况)

In [44]:

```
### TODO: 编写题目要求的函数
def get_index(mat, value=0):
    for i in range(rows):
        for j in range(cols):
            if matrix[i][j] == value:
                return i, j
```

In [45]:

```
print(get_index(matrix))
```

(0, 0)

In [46]:

```
print(get_index(matrix, value = 6))
```

(2, 0)

练习1

请写一个函数, 函数名: find_max,函数有一个必须参数dicts (是一个字典, 请务必保证上面练习题loc字典有正确生成并运行过, 测试时调用函数要用), 该函数返回参数dicts中value最大的键值。

In [47]:

```
### TODO: 编写题目要求的函数
def find_max(dict):
    max_key = None
    max_value = float('-inf')
    for k, v in dict.items():
        if v > max_value:
            max_key = k
    return max_key
```

In [48]:

```
print(find_max(loc))
```

sh

Task 3: 异常处理

问答题

问题:

请描述下

```
try:
    <语句0>
except <名字>:
    <语句1>
else:
    <语句3>
finally:
    <语句4>
```

的运行流程。

回答:

练习

使用异常处理语句，编写一个函数change，该函数有一个输入mat，mat为一个二维列表，函数的功能是将列表中的值是奇数的项置为1，将列表中的值是偶数的项置为0，如果是非数值型内容的项置为字符串error，最后返回这个更改后的字符串。

In []:

```
### Todo
```

In []:

```
print(change(matrix))
```

思考题：综合

任务描述

复习所有语法，解答此思考题

练习

编写一个函数，输入n为偶数时，调用函数求 $1/2+1/4+\dots+1/n$ ，当输入n为奇数时，调用函数求 $1/1+1/3+\dots+1/n$ ，如果是非数值型参数时输出"Error parameter!"，最后输出相加的结果（如果是非数值型参数不输出）。函数名和参数名不做限制。

提示：

循环判断和异常处理的组合应用

In []:

```
### TODO: （此处应有TODO内容）
```

注意: 当你写完了所有的代码，并且回答了所有的问题。你就可以把你的 iPython Notebook 导出成 HTML 文件。你可以在菜单栏，这样导出File -> Download as -> HTML (.html)把这个 HTML 和这个 iPython notebook 一起做为你的作业提交给助教。