数据预处理 (Data Preprocessing)

第7周 Python实习实践手册

Shùjù yùchǔlǐ Dì qī zhōu Python shíxí shíjiàn shǒucè

字符串及文本处理 (String & Text Processing)

开发环境: PyCharm IDE | 编程语言: Python 3.x

kāifā huánjìng: PyCharm IDE | biānchéng yǔyán: Python 3.x

实习一：环境准备与基础导入 (Lab 1: Setup & Import)

Shíxí yī: Huánjìng zhǔnbèi yǔ jīchǔ dǎorù

教授说明 (jiàoshòu shuōmíng): 同学们，今天我们要进行第7周的Python实习。

Tóngxuémen, jīntiān wǒmen yào jìnxíng dì qī zhōu de Python shíxí.

首先，请打开PyCharm，创建一个新的Python文件。

Shǒuxiān, qǐng dǎkāi PyCharm, chuàngjiàn yīgè xīn de Python wénjiàn.

Step 1: 打开PyCharm → 新建项目(New Project) → 创建Python文件

文件名 (File Name): `lab7_string_basics.py`

Step 2: 输入以下导入代码 (Import Libraries)

```
# ============================================
# 第7周实习：字符串及文本处理
# 文件名：lab7_string_basics.py
# ============================================

import pandas as pd
import numpy as np
import re

print('所有库导入成功！ All libraries imported!')
print('Python版本确认完毕，开始实习！')
```

>>> 预期输出 (Expected Output):

```
所有库导入成功！ All libraries imported!
Python版本确认完毕，开始实习！
```

教授提示 (Tips): 如果出现ModuleNotFoundError，请在PyCharm终端(Terminal)中输入: pip install pandas numpy

Rúguǒ chūxiàn ModuleNotFoundError, qǐng zài PyCharm zhōngduān zhōng shūrù: pip install pandas numpy

师生互动 (Interactive): 请大家运行这段代码，看看是否成功。成功的同学请举手！

Qǐng dàjiā yùnxíng zhè duàn dàimǎ, kànkan shìfǒu chénggōng. Chénggōng de tóngxué qǐng jǔshǒu!

实习二：Python字符串基础操作 (Lab 2: String Basics)

Shíxí èr: Python zìfú chuàn jīchǔ cāozuò

教授说明: 现在我们来学习Python字符串(String)的基本操作。字符串是数据预处理(Data Preprocessing)中最常见的数据类型之一。

Xiànzài wǒmen lái xuéxí Python zìfú chuàn de jīběn cāozuò. Zìfú chuàn shì shùjù yùchǔlǐ zhōng zuì chángjiàn de shùjù lèixíng zhī yī.

2.1 字符串创建与基本操作 (String Creation & Basic Operations)

继续在文件 `lab7_string_basics.py` 中添加代码:

```
# ============================
# 2.1 字符串创建与基本操作
# ============================

# 创建字符串
text1 = 'Hello, 数据预处理!'
text2 = "Python字符串处理"
text3 = '''这是
多行字符串'''

print('--- 字符串创建 ---')
print('text1:', text1)
print('text2:', text2)
print('text3:', text3)

# 字符串长度 (Length)
print('\n--- 字符串长度 ---')
print('text1的长度:', len(text1))

# 索引 (Indexing) - 从0开始
print('\n--- 索引操作 ---')
print('第一个字符:', text1[0])    # H
print('最后一个字符:', text1[-1])  # !

# 切片 (Slicing)
```

```
print('\n--- 切片操作 ---')
print('前5个字符:', text1[0:5])   # Hello
print('第7个以后:', text1[7:])    # 数据预处理!
```

>>> 预期输出 (Expected Output):

```
--- 字符串创建 ---
text1: Hello, 数据预处理!
text2: Python字符串处理
text3: 这是
多行字符串


--- 字符串长度 ---
text1的长度: 13


--- 索引操作 ---
第一个字符: H
最后一个字符: !


--- 切片操作 ---
前5个字符: Hello
第7个以后: 数据预处理!
```

教授讲解: 同学们注意，Python中索引(Index)是从0开始的，不是从1开始。这是很多初学者容易犯的错误。

Tóngxuémen zhùyì, Python zhōng suǒyǐn shì cóng 0 kāishǐ de, bú shì cóng 1 kāishǐ.

Zhè shì hěn duō chūxuézhě róngyì fàn de cuòwù.

2.2 常用字符串方法 (Common String Methods)

教授说明: Python提供了很多内置的字符串方法(Built-in String Methods)。我们来逐一练习。

Python tígōngle hěn duō nèizhì de zìfú chuàn fāngfǎ. Wǒmen lái zhúyī liànxí.

```
# ============================
# 2.2 常用字符串方法
# ============================

sample = '  Hello, Data Preprocessing World!  '

# 大小写转换 (Case Conversion)
print('--- 大小写转换 ---')
print('大写(upper):', sample.upper())
print('小写(lower):', sample.lower())
print('首字母大写(title):', sample.title())

# 去除空白 (Strip Whitespace)
print('\n--- 去除空白 ---')
print('原始字符串: [' + sample + ']')
print('去两端空白(strip): [' + sample.strip() + ']')
print('去左空白(lstrip): [' + sample.lstrip() + ']')
print('去右空白(rstrip): [' + sample.rstrip() + ']')

# 查找与替换 (Find & Replace)
print('\n--- 查找与替换 ---')
clean = sample.strip()
print('查找Data位置:', clean.find('Data'))
print('替换:', clean.replace('World', 'China'))

# 分割与连接 (Split & Join)
print('\n--- 分割与连接 ---')
words = clean.split(',')
print('按逗号分割:', words)
rejoined = ' | '.join(words)
print('用|连接:', rejoined)
```

>>> 预期输出 (Expected Output):

```
--- 大小写转换 ---
大写(upper):   HELLO, DATA PREPROCESSING WORLD!
小写(lower):   hello, data preprocessing world!
首字母大写(title):   Hello, Data Preprocessing World!

--- 去除空白 ---
原始字符串: [  Hello, Data Preprocessing World!  ]
去两端空白(strip): [Hello, Data Preprocessing World!]

--- 查找与替换 ---
查找Data位置: 7
替换: Hello, Data Preprocessing China!

--- 分割与连接 ---
按逗号分割: ['Hello', ' Data Preprocessing World!']
```

师生互动 (Interactive): 请大家试试把自己的名字用upper()和lower()转换，看看效果如何?

Qǐng dàjiā shìshi bǎ zìjǐ de míngzì yòng upper() hé lower() zhuǎnhuàn, kànkan xiàoguǒ rúhé?

实习三：Pandas中的字符串处理 (Lab 3: Pandas .str Methods)

Shíxí sān: Pandas zhōng de zìfú chuàn chǔlǐ

教授说明: 在实际数据分析(Data Analysis)中，我们经常需要对DataFrame中的整列数据进行字符串操作。Pandas提供了.str访问器(Accessor)来实现批量字符串处理。

Zài shíjì shùjù fēnxī zhōng, wǒmen jīngcháng xūyào duì DataFrame zhōng de zhěng liè shùjù jìnxíng zìfú chuàn cāozuò.

新建文件 (New File): `lab7_pandas_str.py`

3.1 创建示例数据 (Create Sample Data)

```
# ============================================
# 第7周实习: Pandas字符串处理
# 文件名: lab7_pandas_str.py
# ============================================

import pandas as pd
import numpy as np

# 创建模拟学生信息数据
data = {
    'name': ['  张三  ', 'LI Si', 'wang WU', '  赵六', 'QIAN qi  '],
    'email': ['zhangsan@email.com', 'LISI@Email.COM', 'wangwu@EMAIL.com',
              'zhaoliu@email.com', 'qianqi@Email.Com'],
    'phone': ['138-1234-5678', '139.2345.6789', '137 3456 7890',
              '13612345678', '135-4567-8901'],
    'course': ['大数据分析 Data Analysis', '数据预处理 Data Preprocessing',
               '机器学习 Machine Learning', '大数据分析 data analysis',
               '数据预处理 Data preprocessing']
}

df = pd.DataFrame(data)
print('=== 原始数据 ===')
print(df)
print('\n数据类型:')
print(df.dtypes)
```

>>> 预期输出 (Expected Output):

```
=== 原始数据 ===
       name              email           phone                            course
0    张三    zhangsan@email.com  138-1234-5678        大数据分析 Data Analysis
1     LI Si    LISI@Email.COM  139.2345.6789  数据预处理 Data Preprocessing
2   wang WU  wangwu@EMAIL.com  137 3456 7890    机器学习 Machine Learning
3      赵六  zhaoliu@email.com    13612345678      大数据分析 data analysis
4  QIAN qi   qianqi@Email.Com  135-4567-8901  数据预处理 Data preprocessing
```

教授讲解: 同学们看，这个数据有很多问题——有些名字有多余的空格(Space)，大小写不统一(Inconsistent Case)，电话号码格式(Format)不一样。这就是我们需要做数据清洗(Data Cleaning)的原因。

Tóngxuémen kàn, zhège shùjù yǒu hěn duō wèntí—yǒuxiē míngzì yǒu duōyú de kōnggé, dàxiǎoxiě bù tǒngyī, diànhuà hàomǎ géshì bù yīyàng.

3.2 使用.str方法清洗数据 (Data Cleaning with .str)

```
# ============================
# 3.2 .str方法清洗数据
# ============================


# (1) 去除空格 (Strip Whitespace)
df['name_clean'] = df['name'].str.strip()
print('--- 去除空格后的姓名 ---')
print(df[['name', 'name_clean']])


# (2) 统一大小写 (Standardize Case)
df['email_clean'] = df['email'].str.lower()
print('\n--- 统一小写后的邮箱 ---')
print(df[['email', 'email_clean']])


# (3) 字符串替换 - 统一电话格式 (Standardize Phone Format)
df['phone_clean'] = df['phone'].str.replace('-', '', regex=False)
df['phone_clean'] = df['phone_clean'].str.replace('.', '', regex=False)
df['phone_clean'] = df['phone_clean'].str.replace(' ', '', regex=False)
print('\n--- 统一电话号码格式 ---')
print(df[['phone', 'phone_clean']])


# (4) 字符串包含判断 (String Contains)
df['is_data_course'] = df['course'].str.lower().str.contains('data')
print('\n--- 是否包含Data相关课程 ---')
print(df[['course', 'is_data_course']])
```

教授提示 (Tips): 注意.str方法可以链式调用(Chain Call)，比如 .str.lower().str.contains() 先转小写再判断包含。

Zhùyì .str fāngfǎ kěyǐ liànshì diàoyòng, bǐrú .str.lower().str.contains() xiān zhuǎn xiǎoxiě zài pànduàn bāohán.

师生互动: 请大家思考一下，为什么在判断课程(Course)是否包含'data'之前，要先用.str.lower()转成小写?

Qǐng dàjiā sīkǎo yīxià, wèishénme zài pànduàn kèchéng shìfǒu bāohán 'data' zhīqián, yào xiān yòng .str.lower() zhuǎn chéng xiǎoxiě?

3.3 字符串提取与分割 (String Extraction & Split)

继续在 `lab7_pandas_str.py` 中添加:

```
# ============================
# 3.3 字符串提取与分割
# ============================

# (1) 提取邮箱用户名 (Extract Username from Email)
df['username'] = df['email_clean'].str.split('@').str[0]
print('--- 提取邮箱用户名 ---')
print(df[['email_clean', 'username']])

# (2) 提取邮箱域名 (Extract Domain)
df['domain'] = df['email_clean'].str.split('@').str[1]
print('\n--- 提取邮箱域名 ---')
print(df[['email_clean', 'domain']])

# (3) 字符串长度统计 (String Length)
df['name_length'] = df['name_clean'].str.len()
print('\n--- 姓名字符长度 ---')
print(df[['name_clean', 'name_length']])

# (4) 课程中英文分离 (Separate Chinese & English)
df['course_cn'] = df['course'].str.split(' ').str[0]
print('\n--- 课程中文名 ---')
print(df[['course', 'course_cn']])
```

>>> 预期输出 (Expected Output):

```
--- 提取邮箱用户名 ---
         email_clean  username
0  zhangsan@email.com  zhangsan
1    lisi@email.com      lisi
2  wangwu@email.com    wangwu
3  zhaoliu@email.com   zhaoliu
4   qianqi@email.com    qianqi
```

教授讲解: 这里我们用.str.split('@')把邮箱按@符号分割，再用.str[0]取第一部分(用户名)，.str[1]取第二部分(域名)。这种技巧在数据清洗(Data Cleaning)中非常实用。

Zhèlǐ wǒmen yòng .str.split('@') bǎ yóuxiāng àn @ fúhào fēngē, zài yòng .str[0] qǔ dì yī bùfen, .str[1] qǔ dì èr bùfen.

实习四：正则表达式基础 (Lab 4: Regular Expressions)

Shíxí sì: Zhèngzé biǎodáshì jīchǔ

教授说明: 正则表达式(Regular Expression, 简称regex)是一种强大的文本模式匹配(Pattern Matching)工具。虽然看起来有点复杂，但只要掌握几个基本符号就够用了。

Zhèngzé biǎodáshì shì yīzhǒng qiángdà de wénběn móshì pǐpèi gōngjù.

Suīrán kàn qǐlái yǒudiǎn fùzá, dàn zhǐyào zhǎngwò jǐ gè jīběn fúhào jiù gòu yòng le.

新建文件: `lab7_regex.py`

4.1 正则表达式常用符号速查表 (Regex Cheat Sheet)

符号 (Symbol)	含义 (Meaning)	示例 (Example)	匹配结果 (Match)
`\d`	数字 (Digit, 0-9)	`\d+`	123, 456
`\w`	字母/数字/下划线	`\w+`	hello, abc_123
`\s`	空白字符 (Space)	`\s+`	空格, Tab
`.`	任意字符 (Any)	`a.c`	abc, a1c, a-c
`*`	0次或多次	`ab*c`	ac, abc, abbc
`+`	1次或多次	`ab+c`	abc, abbc (不匹配ac)
`?`	0次或1次	`colou?r`	color, colour
`[]`	字符集合	`[aeiou]`	匹配任一元音
`^`	开头 (Start)	`^Hello`	Hello开头的文本
`$`	结尾 (End)	`world$`	world结尾的文本

教授提示: 不需要记住所有符号！最常用的是 \d(数字)、\w(单词)、\s(空白)、+(一个以上)

Bù xūyào jìzhù suǒyǒu fúhào! Zuì chángyòng de shì \d, \w, \s, +

4.2 正则表达式实践 (Regex Practice)

```
# ============================================
# 第7周实习：正则表达式
# 文件名：lab7_regex.py
# ============================================

import re
import pandas as pd

# === 练习1：从文本中提取电话号码 ===
text = '''
客户张三的电话是138-1234-5678，
李四的号码是139.2345.6789，
王五联系方式：13734567890
'''

# 使用正则表达式查找所有电话号码
pattern = r'1[3-9]\d[-.\s]?\d{4}[-.\s]?\d{4}'
phones = re.findall(pattern, text)

print('=== 练习1：提取电话号码 ===')
print('找到的电话号码：')
for i, phone in enumerate(phones, 1):
    print(f'  {i}. {phone}')
```

>>> 预期输出 (Expected Output):

```
=== 练习1：提取电话号码 ===
找到的电话号码：
  1. 138-1234-5678
  2. 139.2345.6789
  3. 13734567890
```

教授讲解: 我们来分析这个正则表达式(Regular Expression)模式(Pattern)：

`r'1[3-9]\d[-.\s]?\d{4}[-.\s]?\d{4}'` 的含义是：

• 1[3-9] → 以1开头，第二位是3到9之间的数字

• \d → 第三位是任意数字

• [-.\s]? → 可能有分隔符(-, ., 空格)，也可能没有

• \d{4} → 连续4个数字

Wǒmen lái fēnxī zhège zhèngzé biǎodáshì móshì.

4.3 提取邮箱地址 (Extract Email Addresses)

继续在 `lab7_regex.py` 中添加:

```
# === 练习2: 提取邮箱地址 ===
text2 = '''
请联系我们:
技术支持: support@company.com
客服邮箱: service@shop.cn
投诉建议: feedback@company.org
无效格式: not-an-email, @missing.com
'''

email_pattern = r'[\w.+-]+@[\w-]+\.[a-zA-Z]{2,}'
emails = re.findall(email_pattern, text2)

print('\n=== 练习2: 提取邮箱地址 ===')
print('找到的邮箱地址:')
for i, email in enumerate(emails, 1):
    print(f'  {i}. {email}')

# === 练习3: 文本清洗 - 去除特殊字符 ===
messy_text = '这是一段!!很##乱的@@文本，需要&&清洗...'
clean_text = re.sub(r'[!@#$%^&*]+', '', messy_text)

print('\n=== 练习3: 去除特殊字符 ===')
print('清洗前:', messy_text)
print('清洗后:', clean_text)
```

>>> 预期输出 (Expected Output):

```
=== 练习2: 提取邮箱地址 ===
找到的邮箱地址:
  1. support@company.com
  2. service@shop.cn
  3. feedback@company.org

=== 练习3: 去除特殊字符 ===
清洗前: 这是一段!!很##乱的@@文本，需要&&清洗...
清洗后: 这是一段很乱的文本，需要清洗...
```

师生互动: 请大家动手尝试修改正则表达式，试试能不能也把中文标点符号(Punctuation)去

掉？提示：可以加上'，。！'到字符集合[]里面。

Qǐng dàjiā dòngshǒu chángshì xiūgǎi zhèngzé biǎodáshì, shìshi néng bù néng yě bǎ zhōngwén biāodiǎn fúhào qùdiào?

实习五：综合实践项目 (Lab 5: Comprehensive Practice)

Shíxí wǔ: Zōnghé shíjiàn xiàngmù

教授说明: 现在我们把前面学到的所有知识综合运用(Comprehensive Application)。我们来处理一个模拟的新闻标题数据集(News Headlines Dataset)。

Xiànzài wǒmen bǎ qiánmiàn xué dào de suǒyǒu zhīshì zōnghé yùnyòng.

新建文件: `lab7_comprehensive.py`

5.1 创建模拟数据集 (Create Mock Dataset)

```
# ============================================
# 第7周实习: 综合实践
# 文件名: lab7_comprehensive.py
# ============================================

import pandas as pd
import numpy as np
import re

# 创建模拟新闻标题数据
news_data = {
    'headline': [
        '  Breaking: 上海GDP增长8.5%！  ',
        'TECH NEWS: 华为发布新芯片Kirin 9100  ',
        '天气预报：北京明天最高温度32°C!!!',
        '体育: 中国队3:1击败日本队@世界杯',
        '  经济：CPI同比上涨2.3%, PPI下降1.2%  ',
        'Entertainment: 新电影《流浪地球3》票房突破50亿!!',
        '科技: AI人工智能ChatGPT用户突破1亿人   ',
        '健康: 专家建议每天步行10000步以上  ',
    ],
    'category': ['economy', 'TECH', 'Weather', 'sports',
                 'ECONOMY', 'entertainment', 'Tech', 'HEALTH'],
    'date': ['2024-03-01', '2024-03-01', '2024-03-02', '2024-03-02',
             '2024-03-03', '2024-03-03', '2024-03-04', '2024-03-04']
}

df = pd.DataFrame(news_data)
```

```
print('=== 原始新闻数据 ===')
print(df)
print(f'\n数据行数: {len(df)}')
```

教授讲解: 同学们注意看，这个数据集有很多问题需要清洗：

① 标题(Headline)前后有多余空格

② 分类(Category)大小写不统一

③ 标题中有重复的标点符号(Punctuation)

④ 包含特殊字符(@)

Tóngxuémen zhùyì kàn, zhège shùjù jí yǒu hěn duō wèntí xūyào qīngxǐ.

5.2 完整的数据清洗流程 (Complete Cleaning Pipeline)

教授说明: 现在我们一步一步来做数据清洗(Data Cleaning Pipeline)。请跟着一起操作。

Xiànzài wǒmen yī bù yī bù lái zuò shùjù qīngxǐ. Qǐng gēnzhe yīqǐ cāozuò.

```
# ============================
# 5.2 完整数据清洗流程
# ============================

# Step 1: 去除前后空格 (Strip Whitespace)
df['headline_clean'] = df['headline'].str.strip()
print('\n=== Step 1: 去除空格 ===')
print(df['headline_clean'])

# Step 2: 统一分类名称 (Standardize Category)
df['category_clean'] = df['category'].str.lower().str.strip()
print('\n=== Step 2: 统一分类 ===')
print(df['category_clean'].value_counts())

# Step 3: 去除重复标点和特殊字符 (Remove Extra Punctuation)
def clean_headline(text):
    # 去除连续的感叹号
    text = re.sub(r'!{2,}', '!', text)
    # 去除@符号
    text = re.sub(r'@', '', text)
    # 去除多余的空格
    text = re.sub(r'\s+', ' ', text)
    return text.strip()

df['headline_clean'] = df['headline_clean'].apply(clean_headline)
print('\n=== Step 3: 清洗标题 ===')
for i, row in df.iterrows():
    print(f'原始: {row["headline"].strip()}')
    print(f'清洗: {row["headline_clean"]}')
    print('---')
```

教授提示 (Tips): 在这里我们定义了一个自定义函数(Custom Function) clean_headline()，然后用.apply()方法把它应用到每一行数据上。这是Pandas中非常常用的技巧。

Zài zhèlǐ wǒmen dìngyì le yīgè zìdìngyì hánshù clean_headline(), ránhòu yòng .apply() fāngfǎ bǎ tā yìngyòng dào měi yī háng shùjù shàng.

5.3 从新闻标题中提取信息 (Extract Info from Headlines)

```
# ============================
# 5.3 信息提取
# ============================


# (1) 提取数字 (Extract Numbers)
df['numbers'] = df['headline_clean'].apply(
    lambda x: re.findall(r'\d+\.?\d*', x)
)
print('=== 提取到的数字 ===')
for i, row in df.iterrows():
    if row['numbers']:
        print(f'{row["headline_clean"][:30]}...')
        print(f'  → 数字: {row["numbers"]}')


# (2) 提取英文单词 (Extract English Words)
df['english_words'] = df['headline_clean'].apply(
    lambda x: re.findall(r'[A-Za-z]+', x)
)
print('\n=== 提取到的英文单词 ===')
for i, row in df.iterrows():
    if row['english_words']:
        print(f'{row["headline_clean"][:30]}...')
        print(f'  → 英文: {row["english_words"]}')


# (3) 统计标题长度 (Title Length Statistics)
df['title_length'] = df['headline_clean'].str.len()
print('\n=== 标题长度统计 ===')
print(df[['headline_clean', 'title_length']].to_string())
print(f'\n平均长度: {df["title_length"].mean():.1f}')
print(f'最长标题: {df["title_length"].max()}')
print(f'最短标题: {df["title_length"].min()}')
```

教授讲解: 这里我们用到了lambda表达式(Lambda Expression)，它是一种简短的匿名函数(Anonymous Function)写法。re.findall()会返回所有匹配的结果组成的列表(List)。

Zhèlǐ wǒmen yòng dào le lambda biǎodáshì, tā shì yīzhǒng jiǎnduǎn de nìmíng hánshù xiěfǎ.

实习六：课堂自主练习 (Lab 6: Self Practice)

Shíxí liù: Kètáng zìzhǔ liànxí

教授说明: 现在请大家独立完成以下练习。你们有20分钟的时间。遇到问题可以先看前面的代码例子，也可以问旁边的同学或者举手问我。

Xiànzài qǐng dàjiā dúlì wánchéng yǐxià liànxí. Nǐmen yǒu èrshí fēnzhōng de shíjiān.

Yù dào wèntí kěyǐ xiān kàn qiánmiàn de dàimǎ lìzi, yě kěyǐ wèn pángbiān de tóngxué huòzhě jǔshǒu wèn wǒ.

新建文件: `lab7_exercise.py`

练习题 (Exercise) 1: 创建以下数据框并完成清洗

```
# === 练习1: 清洗产品评论数据 ===
import pandas as pd
import re

reviews = pd.DataFrame({
    'comment': [
        '  这个产品非常好!!!推荐购买!!!  ',
        'VERY GOOD product, 质量不错。  ',
        '发货太慢了...等了10天才到...',
        '  价格$99.9很合理, 性价比高  ',
        '客服态度差@@打了3次电话才解决'
    ]
})

# 任务:
# (1) 去除前后空格
# (2) 将连续的标点符号(!或.)替换为单个
# (3) 去除@符号
# (4) 提取评论中出现的数字
# (5) 计算每条评论的字符长度
# (6) 判断评论是否包含'好'或'推荐'


# ===请在下面写你的代码=== #
```

练习题 (Exercise) 2: 从以下文本中提取所有邮箱和电话号码

```
# === 练习2: 信息提取 ===
contact_info = '''
公司通讯录:
张经理 - zhang@company.com - 13812345678
李主任 - li_zhuren@org.cn - 139-8765-4321
王秘书 - wang.mishu@dept.gov.cn - 137.1111.2222
客服热线: 400-888-9999
'''


# 任务:
# (1) 提取所有邮箱地址
# (2) 提取所有手机号码
# (3) 提取所有人名(提示: X经理/主任/秘书)


# ===请在下面写你的代码=== #
```

练习参考答案 (Reference Answers)

> **重要提示:** 请先自己尝试完成练习，再看参考答案!

Qǐng xiān zìjǐ chángshì wánchéng liànxí, zài kàn cānkǎo dá'àn!

练习1 参考答案:

```
# 练习1 参考答案
# (1) 去除空格
reviews['clean'] = reviews['comment'].str.strip()

# (2) 替换连续标点
reviews['clean'] = reviews['clean'].apply(
    lambda x: re.sub(r'!{2,}', '!', re.sub(r'\.{2,}', '.', x))
)

# (3) 去除@
reviews['clean'] = reviews['clean'].str.replace('@', '', regex=False)

# (4) 提取数字
reviews['numbers'] = reviews['clean'].apply(
    lambda x: re.findall(r'\d+\.?\d*', x)
)

# (5) 字符长度
reviews['length'] = reviews['clean'].str.len()

# (6) 包含'好'或'推荐'
reviews['positive'] = reviews['clean'].str.contains('好|推荐')

print(reviews[['clean', 'numbers', 'length', 'positive']])
```

练习2 参考答案:

```
# 练习2 参考答案
# (1) 提取邮箱
emails = re.findall(r'[\w.+-]+@[\w-]+\.[a-zA-Z.]{2,}', contact_info)
print('邮箱:', emails)

# (2) 提取手机号
phones = re.findall(r'1[3-9]\d[-.\s]?\d{4}[-.\s]?\d{4}', contact_info)
```

```
print('手机:', phones)


# (3) 提取人名 (简单方式)
names = re.findall(r'(\w)(?:经理|主任|秘书)', contact_info)
print('人名:', names)
```

实习总结与回顾 (Lab Summary & Review)

Shíxí zǒngjié yǔ huígù

教授总结: 同学们辛苦了！今天我们学了很多内容，让我们一起来回顾一下。

Tóngxuémen xīnkǔ le! Jīntiān wǒmen xué le hěn duō nèiróng, ràng wǒmen yīqǐ lái huígù yīxià.

实习内容	主要知识点 (Key Points)	核心代码 (Core Code)
实习一	环境准备与库导入	`import pandas, re`
实习二	字符串创建、索引、切片	`s[0], s[0:5], len(s)`
实习三	Pandas .str方法	`.str.strip(), .str.lower()`
实习四	正则表达式基础	`re.findall(), re.sub()`
实习五	综合数据清洗	`apply() + 自定义函数`
实习六	课堂独立练习	综合运用以上知识

今日收获 (What We Learned Today)

1. Python字符串(String)是不可变(Immutable)的，操作后会返回新字符串
2. Pandas的.str访问器(Accessor)可以对整列数据批量处理字符串
3. 正则表达式(Regular Expression)是强大的模式匹配(Pattern Matching)工具
4. re.findall()用于提取(Extract)，re.sub()用于替换(Replace)
5. .apply()方法可以将自定义函数(Custom Function)应用到DataFrame的每一行

课后作业 (Homework)

请完成以下作业，下周上课前提交：

作业1: 找一个真实的中文文本数据(比如新闻网站的标题)，创建DataFrame并完成清洗

作业2: 用正则表达式从一段包含联系信息的文本中提取邮箱、电话和网址

Qǐng wánchéng yǐxià zuòyè, xià zhōu shàngkè qián tíjiāo.

下次预告 (Next Class): 第8周 — 日期型数据预处理 (Date Data Preprocessing)

Dì bā zhōu — rìqí xíng shùjù yùchǔlǐ

教授结束语: 今天的实习到这里结束了。大家做得很好！有任何问题可以课后来找我。下课！

Jīntiān de shíxí dào zhèlǐ jiéshù le. Dàjiā zuò de hěn hǎo! Yǒu rènhé wèntí kěyǐ kè hòu lái zhǎo wǒ. Xiàkè!