

# pandas && 数据读写

## 安装

```
pip install pandas -i https://pypi.tuna.tsinghua.edu.cn/simple

# excel
pip install openpyxl -i https://pypi.tuna.tsinghua.edu.cn/simple
pip install xlswriter -i https://pypi.tuna.tsinghua.edu.cn/simple
```

## 数据读写

### CSV,JSON,Excel,mysql

[https://pandas.pydata.org/pandas-docs/stable/user\\_guide/io.html](https://pandas.pydata.org/pandas-docs/stable/user_guide/io.html)

api

- read\_csv , read\_json , read\_excel,read\_sql
- to\_csv , to\_json, to\_excel , to\_sql

### mysql

```
pip install pandas -i https://pypi.tuna.tsinghua.edu.cn/simple
pip install sqlalchemy -i https://pypi.tuna.tsinghua.edu.cn/simple
```

```
# pandas
import pandas as pd

# SQLAlchemy
from sqlalchemy import create_engine
```

```
# 连接
if True:
    engine = create_engine(
        "mysql+pymysql://root:password@192.168.0.10:33318/renren_cloud_basic"
    )
    query = "SELECT * FROM park_user"
    # chunksize=1000000
    df = pd.read_sql(query, engine)
    # 打印信息
    print(df.info())
    # 打印列名
    print(df.columns)
```

## 使用mysqlconnector

- read\_sql\_table(table\_name,connection)
- read\_sql\_query(sql,connection)
- read\_sql(sql,connection)

## 打印信息

- 打印信息

```
row_size, col_size = grouped.shape
log.info(f"row_size:{row_size},col_size:{col_size}")
```

```
# row_size:13,col_size:7
```

- 打印信息

```
# 打印信息
dataFrame["year"] = pd.to_datetime(dataFrame["date"]).dt.year
```

- 打印信息
- 打印信息
- 打印信息

- `df.groupby(["country", "continent", "year"]).agg({`
- `df.groupby(["country", "continent", "year"]).agg({`

```
grouped = (
    dataframe.groupby(["country", "continent", "year"])
    .agg(
        {
            "country": "sum",
            "continent": "sum",
            "year": "sum",
            "value": "sum",
        }
    )
    .reset_index() # reset index
)

# Print the grouped data
grouped.index.name = "year"
```

- `df.groupby(["country", "continent", "year"]).agg({`

```
# Print the grouped data
print(grouped["country"])

# Print the grouped data
print(grouped[grouped["country"] == "USA"])
```

## ExcelWriter

pandas.ExcelWriter

```
# Create an ExcelWriter object
with pd.ExcelWriter("./files/output.xlsx", engine="xlsxwriter") as writer:

    # Write the grouped data to the Excel file
```

```
# grouped.to_excel(writer, sheet_name="Sheet1", startrow=1, startcol=1, index=True)
```

# 

--	--	--	--	--

```
workbook = writer.book
```

```
worksheet = None
```

```
# worksheet
```

```
if "Sheet1" in writer.sheets:
```

```
worksheet = writer.sheets["Sheet1"]
```

```
else:
```

```
worksheet = workbook.add_worksheet("Sheet1")
```

# 

--	--	--	--

```
default_format = {
```

```
"bold": False,
```

# 

--	--

```
"border": 1,
```

# 

--	--	--	--

```
"valign": "vcenter",
```

# 

--	--	--	--

```
"text_wrap": True,
```

# 

--	--	--	--

```
# "align": "center",
```

# 

--	--	--	--

```
"font_size": 9,
```

}

# 

--	--	--	--

```
cell_format_default = workbook.add_format(default_format)
```

```
cell_format_head = workbook.add_format(
```

{

```
**default_format,
```

$$**\{$$

```
"align": "center",
```

```
"bold": True,
```

 $\},$ 

}

)

[illegible]

```
worksheet.set_column("A: A", 10)
```

```
worksheet.set_column("B:B", 30)
```

```
worksheet.set_column("E: H", 12)
```

```
worksheet.insert_image(
```

```
". /files/logo.png",
```

```
"x_scale": first_row_height / logo_image_height,
```

```
"y_scale": first_row_height / logo_image_height,
```

```
"x offset": 5,
```

```
"y_offset": 5,
```

 $\},$ 

)

```
worksheet.merge_range("A1:B1", "", cell_format_head)
```

```
worksheet.merge_range("C1:H1", "#####", cell_format_head)
```

```
worksheet.set_row(0, first_row_height)
```

```
worksheet.write("A2", " ", cell_format head)
```

```
worksheet.write_column("A3", range(1, len(grouped) + 1), cell_format_head)
```

```
row_offset = 2
```

```
col offset = 1
```

```
for row num, row data in grouped.iterrows():
```

```
for col_num, col_data in enumerate(row_data):
```

```
worksheet.write(
```

row num + row offset,

col num + col offset,

col data,

cell format default,

)

```
start row = 0
```

```
for row num, row data in grouped.iterrows():
```

```

current_value = row_data[" "]
next_row = row_num + 1
next_value = (
    grouped.iloc[next_row][" "] if next_row < len(grouped) else None
)
if current_value != next_value:
    log.info(f"{start_row} - {row_num} - {current_value}")
    if row_num - start_row > 0:
        worksheet.merge_range(
            start_row + row_offset,
            grouped.columns.get_loc(" ") + col_offset,
            row_num + row_offset,
            grouped.columns.get_loc(" ") + col_offset,
            current_value,
            cell_format_default,
        )
    start_row = next_row

```

#22

11 2025 19:11:35

25 2025 12:02:51