



AniPython

pandas读excel文件



```
pandas.read_excel(  
    io,                # 路径  
    parse_dates=False, # 指定解析成日期格式的列  
    date_parser=None,  # function, 解析日期格式的函数  
)
```



```
pandas.read_excel(  
    io,                                # 路径  
    parse_dates=False,                # 指定解析成日期格式的列  
    date_parser=None,                 # function, 解析日期格式的函数  
)
```

不需要传 `date_parser` 参数

20200120  
2020/1/20  
1/20/2020  
2020-01-20  
2020-01-20 00:10:10

需要传 `date_parser` 参数

2020年1月20日



# pandas.read\_excel()参数

**parse\_dates**: 解析成日期格式

- **True** -> 尝试解析index
- **[0, 1]** 或 **['a', 'b']** -> 尝试解析指定列作为一个单独的日期列
- **[[0, 1, 2]]** -> 结合多列解析为单个日期列
- **{'日期': [0, 1, 2]}** -> 同上, 结果的列名改为'日期'



# pandas.read\_excel()参数

**parse\_dates**: 解析成日期格式

- **True** -> 尝试解析index
- **[0, 1]** 或 **['a', 'b']** -> 尝试解析指定列作为一个单独的日期列
- **[[0, 1, 2]]** -> 结合多列解析为单个日期列
- **{'日期': [0, 1, 2]}** -> 同上, 结果的列名改为'日期'



# pandas.read\_excel()参数

**parse\_dates:** 解析成日期格式

- True -> 尝试解析index
- [0, 1] 或 ['a', 'b'] -> 尝试解析指定列作为一个单独的日期列
- [[0, 1, 2]] -> 结合多列解析为单个日期列
- {'日期': [0, 1, 2]} -> 同上, 结果的列名改为'日期'



# pandas.read\_excel()参数

**parse\_dates:** 解析成日期格式

- True -> 尝试解析index
- [0, 1] 或 ['a', 'b'] -> 尝试解析指定列作为一个单独的日期列
- [[0, 1, 2]] -> 结合多列解析为单个日期列
- {'日期': [0, 1, 2]} -> 同上, 结果的列名改为'日期'



# pandas.read\_excel()参数

**parse\_dates:** 解析成日期格式

- True -> 尝试解析index
- [0, 1] 或 ['a', 'b'] -> 尝试解析指定列作为一个单独的日期列
- [[0, 1, 2]] -> 结合多列解析为单个日期列
- {'日期': [0, 1, 2]} -> 同上, 结果的列名改为'日期'



# pandas.read\_excel()参数

**parse\_dates**: 解析成日期格式

- **True** -> 尝试解析index
- **[0, 1]** 或 **['a', 'b']** -> 尝试解析指定列作为一个单独的日期列
- **[[0, 1, 2]]** -> 结合多列解析为单个日期列
- **{'日期': [0, 1, 2]}** -> 同上, 结果的列名改为'日期'



# pandas.read\_excel()参数

`date_parser`: 解析日期格式的函数

```
date_parser=lambda x: pd.to_datetime(x, format='%Y年%m月%d日')
```