

Change Value

Áp Dụng Nhiều Hàm với Pipe:	<code>df.pipe(tinh_tong).pipe(bo_cot, " a")</code>
Áp Dụng Hàm Cho Một Cột:	<code>df["col1"] = df[" col 1"].apply(lambda row: row * 2)</code>
Áp Dụng Hàm Cho Từng Phần Tử:	<code>df.applymap(lambda val: " failed " if val < 5 else " passed ")</code>
Gán Giá Trị Cho Nhiều Cột Mới:	<code>df = df.assign(col1= lambda x: x.col1 * 100 + x.col2).assign(col 4=1 lambda x: x.col2 * x.col3)</code>
Thay Đổi Giá Trị của Series Bằng Dictionary:	<code>s.map({"a": 1, " b": 2, " c": 3})</code>
Chuyển Mỗi Phần Tử Trong Iterable Thành Một Dòng:	<code>df.explode("a")</code>
Tách Chuỗi Thành Nhiều Dòng:	<code>df.a = df.a.str.split(" ,") df.explode("a ")</code>
Forward Fill Trong pandas:	<code>df = df.fillna(method="ffill")</code>
Thay Thế Giá Trị Trống Bằng Giá Trị Phổ Biến Nhất:	<code>df.fillna(df.mode().iloc[0])</code>
Mã Hóa Giá Trị Phân Loại Trong DataFrame:	<code>df["encoded_col1"] = df[" col 1"].astype("category").cat.codes</code>
Tối Ưu Hóa Mã với Phép Toán Vector:	<code>df = df.fillna(fill_dict)</code>
Ném Ngoại lệ Cho Gán Giá Trị Liên Kết:	<code>pd.options.mode.chained_assignment = " raise" df[df[" col 1"] > 0][" col 2"] = 1</code>

Truy Cập Dữ Liệu

Truy Cập Nhóm Các Dòng với loc và iloc:	<code>s.iloc[0] s.loc[20]</code>
pd.Series.between: Chọn Các Dòng Chứa Giá Trị Trong Một Phạm Vi:	<code>s[s.between(0, 10)]</code>
pandas.Series.pct_change: Tính Phần Trăm Thay Đổi Giữa Giá Trị Hiện Tại và Trước Đó:	<code>df.a.pct_change()</code>
Tính Sự Khác Biệt Giữa Các Dòng của DataFrame:	<code>df.diff()</code>
DataFrame.diff và DataFrame.shift: Lấy Sự Khác Biệt Giữa Các Dòng Trong Một Cột:	<code>diff = df.diff() shift = diff.shift(-1)</code>
Chuyển pandas Series thành Mảng NumPy:	<code>s.values</code>
df.to_dict: Chuyển DataFrame thành Từ Điển:	<code>df.to_dict()</code>



Truy Cập Dữ Liệu (cont)

Lấy Số Lần Xuất Hiện và Phần Trăm của Một Giá Trị Trong Một Cột:	<code>size.value_counts()</code> <code>size.value_counts(normalize=True)</code>
<code>pandas.DataFrame.corrwith</code> : Tính Tương Quan Giữa 2 DataFrame:	<code>df1.corrwith(df2)</code>
<code>pandas.cut</code> : Chia Giá Trị của DataFrame thành Các Khoảng Rời Rạc:	<code>pd.cut(df["a"], bins=bins)</code>
<code>pandas.qcut</code> : Chia Giá Trị của DataFrame thành Các Khoảng Có Số Lượng Phần Tử Bằng Nhau:	<code>pd.qcut(df["a"], q=3)</code>
<code>DataFrame.cumsum</code> : Lấy Tổng Cộng Tích Lũy qua Mỗi Cột:	<code>df.cumsum()</code>
<code>pandas.DataFrame.cummax</code> : Lấy Giá Trị Tối Đa Tích Lũy:	<code>nums.cummax()</code>
Lấy Tổng Của Tất Cả Cột Trong DataFrame:	<code>df.sum(axis=1)</code> <code>df.sum(axis=0)</code>

Làm Việc với Datetime

Chuyển Đổi Các Cột thành Đối Tượng Thời Gian Khi Sử Dụng pandas để Đọc Các Tập CSV:	<code>pd.read_csv("data.csv", parse_dates=["date_column_1", "date_column_2"])</code>
<code>pandas.DateOffset</code> : Thêm Khoảng Thời Gian vào Một Điểm Thời Gian của pandas:	<code>ts + DateOffset(months=3)</code> <code>ts + DateOffset(years=3, hours=3)</code> <code>ts + BDay(n=6)</code>
<code>DataFrame.rolling</code> : Tính Trung Bình Của n Dữ Liệu Trước Đó Sử Dụng pandas:	<code>df.rolling(3).mean()</code>
<code>pandas.Grouper</code> : Nhóm Giá Trị Dựa trên Một Tần Suất Cụ Thể:	<code>df.groupby(pd.Grouper(key="date", freq="1W")).mean()</code>
<code>pandas.Series.dt</code> : Truy Cập Các Thuộc Tính Thời Gian của Một Series của pandas:	<code>df["date"].dt.year</code> <code>df["date"].dt.time</code>
Lấy Các Dòng trong Một Phạm Vi Năm:	<code>df.loc["2019:"]</code>
<code>pandas.reindex</code> : Thay Thế Giá Trị Của Các Ngày Thiếu Trong Một Dãy Ngày Bằng 0:	<code>new_s = s.reindex(new_index, fill_value=0)</code>
Chọn Các Dòng Của DataFrame Trước hoặc Sau Một Ngày Cụ Thể:	<code>df[df.date <= "2021-07-21"]</code>
<code>resample</code> : Tái Mẫu Dữ Liệu Chuỗi Thời Gian:	<code>s.resample('2D').sum()</code>
Dịch Chỉ Số Của DataFrame Theo Số Kỳ Hạn Cụ Thể:	<code>df.shift(periods=1)</code>

